

Jere Peuhkuri

MODULAARINEN TUOTESUUNITTELU

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Nillo Adlin
Toukokuu 2021

TIIVISTELMÄ

Jere Peuhkuri: Modulaarinen tuotesuunnittelu
Kandidaatintyö
Tampereen yliopisto
Konetekniikka, Teknisten tieteiden koulutus
Toukokuu 2021

Tämän tutkimuksen tarkoituksena oli tutkia modulaaristen tuoterakenteiden suunnittelua. Tutkimusongelmana oli, mitä ovat modulaariset tuoterakenteet, miten modulaarinen tuoterakenne suunnitellaan, minkälaisia lähestymistapoja suunnittelussa on. Tutkimustyötä tehtiin kirjallisuusselvityksen avulla tutkimalla kirjallisuutta modulaaristen tuoterakenteiden ja tuoteperheiden suunnittelusta.

Kirjallisuusselvityksen aikana modulaarinen tuoterakenne määriteltiin sellaiseksi järjestelmäksi, jossa tuote koostuu tietyistä rungosta tai päämoduulista, johon muita moduuleja lisätään mahdollistamaan toiminnallisuuden lisäämistä. Moduulit ovat määriteltyjä sellaisiksi elementeiksi, joilla on selkeät rajapinnat toisiinsa ja ne pystyvät toteuttamaan niille tarkoitetun funktion muista moduuleista riippumatta. Lähestymistapoja tuoterakenteen suunnitteluun havaittiin 5: asiakaskeksinen tuotekehitys, matriisipohjaiset, indeksipohjaiset, optimointipohjaiset ja funktiopohjaiset lähestymistavat. Ne eroavat toisistaan tavalla, jolla ne osittelevat tuotteen moduuleihin. Lisäksi löydettiin 7 hyvän modulaarisen tuoterakenteen suunnittelun periaatetta. Periaatteet ovat: 1. Kehitä moduuleja, jotka voivat suorittaa kokonaisen funktion 2. Kehitä pohja tuotteelle, johon moduulit kiinnittyvät 3. Suunnittele päämoduulit keskenään vaihdettaviksi 4. Varaa tilaa erilaisille moduuleille 5. Määritä alueet, joille muutoksia ei voi tehdä 6. Hyödynnä standardointia siellä, missä asiakas ei odota variaatiota 7. Suunnittele standardoidut paikat ja liitokset lisävarusteille, johdoille ja letkuille.

Työn tuloksena todettiin monien moduulien suunnittelun lähestymistapojen hyödyntävän matriiseja, geneettisiä algoritmeja ja kilpailijoiden vertailua sekä analysointia tuotteen kehityksessä. Tulevaisuudessa tutkimusta voitaisiin jatkaa tarkastelemalla lisää tässä tutkimuksessa käytettyä modularisoinnin lähestymistapoja jokaisesta kategoriasta mahdollisten nyt huomaamatta jääneiden havaitsemiseksi ja työssä löydettyjen hyvän nykyaikaisen modulaarisen tuoterakenteen kehitysprosessin piirteiden vahvistamiseksi. Tutkimus nojasi vahvasti nelijakoiseen luokitukseen lähestymistavoista. Jatkotutkimus voisi tutkia aihetta käyttäen erilaista luokitusperiaatetta modularisoinnin lähestymistavoille, jolloin saadut tulokset voisivat muodostaa erilaisen kuvan kuin nyt.

Avainsanat: Modulaarisuus, modulaariset tuoterakenteet, tuotesuunnittelu

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. MODULAARISET TUOTERAKENTEET	3
2.1 Historia	4
2.2 Modulaarinen tuoterakenteen hyödyntäminen teollisuudessa	5
3. MODULAARISEN TUOTERAKENTEEN SUUNNITTELU	8
3.1 7 Periaatetta modulaaristen rakenteiden suunnitteluun	8
3.2 Asiakaslähtöinen tuotekehitys	11
3.3 Funktiokeskeiset työkalut	14
3.4 Optimointipohjaiset työkalut	18
3.5 Matriisipohjaiset työkalut	20
3.1 Indeksipohjaiset työkalut	24
4. YHTEENVETO	28
LÄHTEET	30

LYHENTEET JA MERKINNÄT

HOQ	Laadun talo
DSM	Design Structure matriisi
PFPF	Tuoteperheen rangaistusfunktio
GVI	General Variety Index
CI	Coupling index
MDL	Lyhyin mahdollinen kuvaus

1. JOHDANTO

Teollisuus on jatkuvassa muutoksessa. Teknologia kehittyy jatkuvasti, samoin kilpailun taso ja asiakkaiden vaatimien tuotteiden vastaavuus heidän tarpeisiinsa. Kasvanut paine luoda asiakkaalle arvoa samalla omia kustannuksia pienentäen on johtanut tuotteiden valmistamiseen panostamiseen (Krause & Eilmus, 2011). Hukkaa vähentämään pyrkivä lean-filosofia sai alkunsa Japanissa Toyotan aloitteesta 1950-luvulla (Gupta & Jain, 2013). Tuotesuunnittelun painopiste 1960-luvulta lähtien kohdistunut valmistettavuuteen. Modulaarinen tuotesuunnittelu on keskittynyt helpompaan valmistettavuuteen, sekä nopeampaan ja kustannustehokkaampaan tuotekehitykseen (Pakkanen et al., 2016).

Modulaarinen tuote koostuu komponenteista, jotka yhdessä toteuttavat tietyn tehtävän eli funktion. Moduuleilla on selkeät rajapinnat eivätkä komponentit yhden moduulin sisällä ole vuorovaikutuksessa toisen moduulien komponenttien kanssa. Modulaariset tuoterakenteet mahdollistavat massaräätälöivät ja asiakkaan toiveiden mukaan koottavat tuotteet. Tuotevariaatiot muodostetaan yhdistämällä halutut moduulit toisiinsa, mikä mahdollistaa tuotteiden myöhästetyn erilaistamisen ja tilausten lyhyemmät läpimenoajat. Viime vuosikymmeninä teollisuudessa on lisääntynyt asiakkaalle arvon tuottamiseen keskittyvä ajattelu. Asiakaskeskeinen tuotteiden räätälöinti lisää asiakkaiden tyytyväisyyttä tuotteeseen ja lojaaliutta yritykseen. Tämä on lisännyt modulaarisen tuoterakenteen suosiota.

Tämän työn tarkoituksena on selvittää kirjallisuuskatsauksen keinoin, mitä ovat modulaariset tuoterakenteet, miten modulaarinen tuoterakenne suunnitellaan, minkälaisia lähestymistapoja suunnittelussa on. Tutkimuksessa käydään läpi Pakkanen et al. (2016) neljä kategoriaa moduulien kehittämiseksi, joka eroavat toisistaan perusteiden, joilla ne osittelevat komponentit moduuleihin. Nämä neljä lähestymistapaa ovat optimointipohjaiset, matriisipohjaiset, funktionaaliset ja indeksipohjaiset. Neljää lähestymistavan edustajia verrataan toisiinsa, jotta löydettäisiin lähestymistavan ydin. Lisäksi työssä esitellään ajatus asiakaskeskeisestä tuotekehityksestä. Näistä viidestä eri lähestymistavasta pyritään etsimään yhteneväisyyksiä ja löytämään peruspiirteitä, joita hyvällä modularisointi lähestymistavalla on oltava. Kirjallisuuskat-

saus keskittyy puhtaasti lähestymistapojen vertailuun niiden kyvyssä löytää optimaalinen ratkaisu modulaariselle tuoterakenteelle ja jättää siis ulkopuolelle varastojen näkökulma ja taloudellinen näkökulma.

Ensimmäisessä luvussa määritellään, mitä on modulaarinen tuoterakenne, minkälaisia erilaisia modulaarisuuden tyyppejä on, sekä mitkä sen hyödyt ovat. Toisessa luvussa esitellään modulaarinen tuotesuunnittelun piirteitä ja työkaluja, joilla moduuleja tunnistetaan tuotteessa.

2. MODULAARISET TUOTERAKENTEET

Fellini et al. (2006) mukaan tuoterakenne kuvaa, miten komponentit liittyvät toisiinsa tai järjestetty tuotteessa. Modulaarinen tuoterakenne tarkoittaa sellaista tilanetta, jossa tuote koostuu tietyistä rungosta tai päämoduulista, johon muita moduuleja lisätään mahdollistamaan toiminnallisuuden lisäämistä. Moduulit ovat määriteltäviä sellaisiksi elementeiksi, joilla on selkeät rajapinnat toisiinsa ja ne pystyvät toteuttamaan niille tarkoitetun funktion muista moduuleista riippumatta.

Modulaarinen tuotearkkitehtuuri eroaa yksittäisten tuotteiden arkkitehtuurissa muutamilla kriittisillä osa-alueilla. Tärkeimpänä erona arkkitehtuurien välillä on yksittäisten tuotteiden osien keskinäisen vuorovaikutusten monimutkaisuus ja päällekkäisyys (Fujimoto, 2007). Hänen mukaansa tuotteen elementtien vaihto tai muuntaminen on vaikeaa ilman muutoksia koko lopputuotteeseen. Kun koko tuotetta joudutaan muokkaamaan, siihen myytävät lisävarusteetkin voidaan joutua suunnittelemaan uudestaan muuttuneiden kiinnityspaikkojen sijainnin, asennon tai varusteille varatun tilan takia (Fujimoto, 2007).

Modulaaristen tuotteiden osat vaikuttavat toisiinsa selvien rajapintojen kautta ja niiden moduulit ja interaktiot ovat tarkasti määriteltäviä. Modulaarinen tuotearkkitehtuuri on suunniteltu vastaamaan tiettyyn variaation tarpeeseen, lisäämään komponenttien samankaltaisuutta, vähentämään kompleksisuutta ja sen vaikutusta toiminnossa ja tarjoamaan hyötyä alueilla, joilla sitä siltä odotetaan. (Andreasen, 2011)

Pine (1993) määritteli modulaarisuuden tyypeiksi component-sharing, component-swapping, cut-to-fit, bus, sectional ja mix modularity. Component-sharing modulaarisuus tarkoittaa komponenttien käyttöä useissa tuotteissa yrityksen tuotelinjastossa (Ramdas & Randall, 2008), kun taas component-swapping modulaarisuus viittaa kykyyn vaihtaa eri komponentteja tuotelinjaston tuotteiden sisällä (Ghaffari & Ulsoy, 2017). Cut-to-fit modulaarisuus viittaa mahdollisuuteen muokata moduulin kokoa tai muotoa asiakkaiden tarpeiden mukaan (Gupta & Okudan, 2007). Bus-modulaariset moduulit on suunniteltu sopimaan yhteen tietyn alustan kanssa (Gupta & Okudan, 2007). Sectional modulaarisuus hyödyntää tiettyjä standardoituja osia, joista voidaan koota erilaisia moduuleja ja kokonaisuuksia erilaisiin tarkoituksiin (Gupta & Okudan, 2007). Mix modulaarisuus tarjoaa mahdollisuuden valita tiettyjen

ennalta määritettyjen vaihtoehtojen väliltä, mutta prosessit ja rajapinnat ovat standardoituja. Lehtosen (2007) mukaan Otto ja Wood (2001) kuvaa mix modularisuutta verkoksi, joka yhdistää elementtejä.

(Kamrani, Nasr, 2010) Mukaan modulaarisuus tuo mukanaan monia etuja. Komponenttien ostohinnat saadaan pienemmiksi, koska niitä ostetaan vähemmän erilaisia ja suurempia määriä tiettyjä komponentteja. Tuotteiden päivitys on helppoa funktionaalisten moduulien ansiosta. Tuotevariaatio saadaan korkeammaksi kohtuullisella määrällä osia. Toimitusajat lyhentyvät ja tuotesuunnittelu ja -testaus helpottuu toisistaan irrallisten tuotefunktioiden ansiosta (Kamrani, Nasr, 2010). Pahl et al. (2007) jatkaa etujen luetteloa lisäämällä etuja valmistajan näkökulmasta. Modulaaristen tuoterakenteiden etuja valmistajan näkökulmasta ovat käytössä olevan dokumentaation määrä projekti suunnitteluun, suunnittelutyötä tarvitaan enää epätavallisiin tilauksiin, joita ei voida ennustaa, aikataulutukset helpottuu, tilausten läpimenoajat lyhenevät, sillä moduuleja voidaan rakentaa samanaikaisesti, tietotekniikan hyödyntäminen tilausten täyttämässä. (Pahl et al., 2007)

Pahl et al. (2007) mainitsee modulaarisilla tuoterakenteilla olevan myös heikkouksia. Esimerkiksi, kun tällainen tuoterakenne otetaan käyttöön, menetetään jonkin verran joustavuutta tuotannossa, eikä kaikkiin asiakastarpeisiin pystytä vastaamaan. Tuotekehitys on suuri panostus, joten tuotesukupolvien välinen aika venyy pitkäksi. Moduulien yhteensovittaminen nostaa laatuvaatimuksia, joka nostaa tuotantokustannuksia. Optimaalisten modulaaristen systeemien määrittäminen voi olla haasteellista, koska on otettava huomioon sekä asiakkaan, että valmistajan intressit. (Pahl et al., 2007)

2.1 Historia

Ennen teollisuus keskittyi valmistamaan standardisoituja tuotteita mahdollisimman suurina määrinä. Simpson et al. (2006) mukaan monilla aloilla tämä pitää paikkaansa vieläkin, mutta yhä useammille asiakkaille eivät enää riitä korkealaatuiset standardiratkaisut. Asiakasvaatimusten lisääntyminen johtuu teollisuuden kehittymisestä ja siitä seuraavasta monimutkaisempien tuotantovälineiden tarpeesta, sekä tarpeesta differentioitua kilpailijoista (Simpson et al., 2006). Toiset kilpailijat valitsevat strategiakseen tuotteiden tuottamisen halvalla. Toiset taas keskittyvät asiakkaiden toiveiden mukaan suunniteltuihin tuotteisiin. Viime vuosikymmeninä on myös noussut ajatus näiden kahden idean yhdistämisestä ja molempien hyötyjen hyödyntämiseksi. Näitä ajatuksia hyödyntäviä valmistusmenetelmiä kutsutaan Configure-to-order ja Assemble-to-order valmistustavoiksi. Niissä molemmissa ajatus keskittyy

siihen, että tarjotaan asiakkaalle hänen toiveidensa mukainen tuote, mutta se valmistetaan valmiiksi kasatuista moduuleista kustannusten vähentämiseksi (Gardner, 2009).

Asiakastilauksen mukaan koottavat ja konfiguroitavat tuotteet koostuvat moduuleista. Moduulien kehittäminen on ollut tutkijoiden tutkimuksen kohteena jo kymmeniä vuosia. Yrityksiä kiinnostaa erityisesti moduulien suunnitteleminen olemassa olevista osista, sillä kokonaan uusien tuotteiden suunnitteleminen on kallista ja riskialtista (Pakkanen et al., 2016). Pakkanen et al. (2016) esittävät tähän tarpeeseen termin ”Design reuse” eli aikaisempien suunnitelmien ja komponenttien hyödyntäminen tuotteen suunnittelussa.

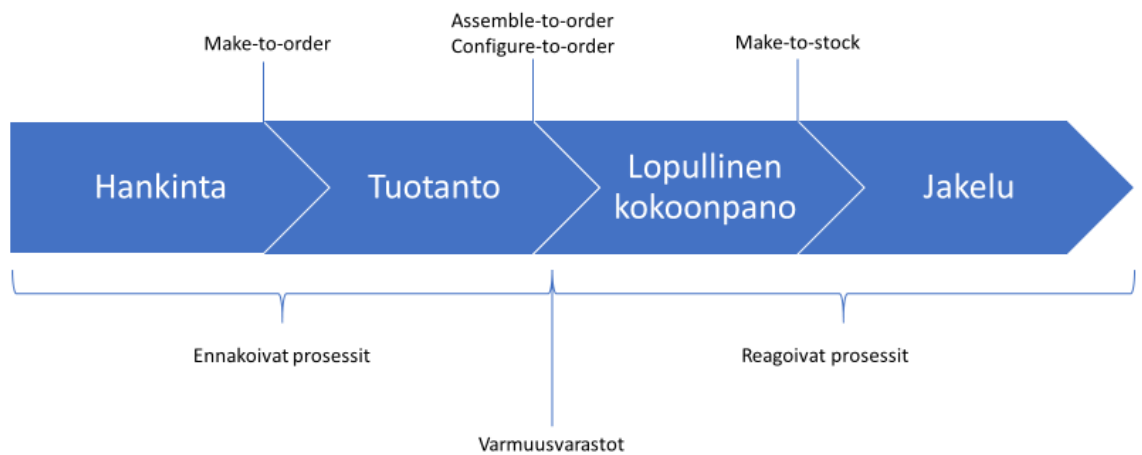
Aikaisempien suunnitelmien hyödyntämisen mahdollistaa modulaaristen tuoteperheiden rakentaminen ja rajapintojen standardisointi. Tuoteperheet eroavat erillisesti suunnitelluista tuotteista niiden jakamalla alustalla (Simpson et al., 2006). Tuoteperheen tuotteet rakennetaan samalle alustalle, josta niihin lisätään tuotteet toisistaan erottavat moduulit (Simpson et al., 2006). Rajapintojen standardisointi taas tarkoittaa moduulien kiinnityskohtien standardisointia muihin moduuleihin tai tuotealustaan (Greve & Krause, 2009).

2.2 Modulaarinen tuoterakenteen hyödyntäminen teollisuudessa

Seiler et al. (2019) mukaan markkinoiden tarpeen yksilöllisille tuotteille aiheuttavan valmistajille korkean tason sisäistä ja ulkoista variaatiota, mikä kasvattaa tuotteiden kompleksisuutta. Tuotteiden kompleksisuuden kasvaessa tuotannon kustannukset nousevat. Kompleksisuuden pienentämiseksi yritykset ovat alkaneet hyödyntää modulaarisia tuoterakenteita, mutta ongelmana on ollut, ettei perinteinen Engineer-to-order prosessi ei pysty vastaamaan modulaaristen tuoterakenteiden käyttöönotossa kohdattaviin haasteisiin. Tähän tarpeeseen on kehitetty Configure-to-order prosessi. (Seiler et al., 2019)

Teollisuudessa käytettävät valmistusprosessit voidaan jakaa karkeasti kolmeen kategoriaan: make-to-stock, configure-to-order ja Make-to-order (Seiler et al., 2019). Make-to-stock tuotanto sopii tuotteille, joilta ei vaadita variaatiota ja joita on tuotettava suuri määrä halvalla. Make-to-stock toimintatavalla tuotetut hyödykkeet odottavat varastossa asiakkaan tilausta. Engineer-to-order kuvaa tämän tilanteen vastaakohtaa, missä jokainen tilaus suunnitellaan asiakkaalle erikseen. Make-to-stock

tuotantomäärät ovat pieniä ja tuotteet kalliita, mutta ne voidaan parhaiten saada vastaamaan asiakkaan tarpeita. Configure-to-order tuotannossa tuotteet kootaan asiakkaan toiveiden mukaisesti valmiiksi määritellyistä moduuleista (Chen-Ritzo et al., 2011). Configure-to-order tuotannolla pyritään tarjoamaan edullisempia tuotteita kuin Make-to-order, ja suurempaa variaatiota kuin Make-to-stock tuotannolla. Kuvassa 1 esitetään, missä vaiheessa erilaisten tuotantotyyppien on huomioitava tilaus toiminnassaan. Esimerkiksi make-to-order tuotannossa tuotteet eroavat toisistaan jo hankintavaiheessa, koska kaikki tilaukset ovat erilaisia ja tuotteet koostuvat eri osista.



Kuva 1: Liiketoimintaprosessit (Seiler et al., 2019)

Modulaariset tuoterakenteet mahdollistavat Cto toimintatavan käytön. Kuluttajatuotteissa ja yleisemmissä teollisuustuotteissa asiakas pääsee itse konfiguroimaan tuotteensa. Asiakas valitsee hänen tarvitsemansa tuoteominaisuudet tai jopa komponentit, joiden perusteella yritys määrittää kokoonpanon. Kokoonpanon koostuessa ennalta suunnitelluista, määritellyistä moduuleista, tuloksena on ennalta hyväksytty tuotevariantti. Moduulit valmistaan itsenäisesti toisistaan omilla osastoillaan, josta ne kuljetetaan kokoonpanolinjastolle valmiiksi tuotteeksi yhdistettäväksi. Cto prosessin etuna on juuri pienet valmiiden tuotteiden varastot, koska moduulit eli osakokoonpanot voidaan säilyttää erillään kokoonpanohetkeen asti. (Seiler et al., 2019)

Suuremmissa projekteissa pidetään projektin aloituspalaveri projektinjohtajan johdolla (Seiler et al., 2019). Projektin johtaja antaa ohjeet osastojen johtajille, jotka aloittavat tuotekehityksen. Kun tuotesuunnittelu on valmis, suunnitelmat lähetetään eri osastoille, jotka voivat valmistaa osat itsenäisesti. Valmiit moduulit kuljetetaan testikoottavaksi ja esikokoonpano asiakkaan tarkistettavaksi. Asiakkaan hyväksytyä tuote, se puretaan ja lähetään asiakkaalle, jossa se kootaan lopullisesti. (Seiler et al., 2019)

Tietokoneteollisuus on ollut konfiguroitavan tuotannon edelläkävijä. Tietokoneet ovat jo pitkään koostuneet joukosta eri valmistajien valmistamia komponentteja, joilla on standardoidut rajapinnat. Asiakkaat voivat internetissä valita haluamansa komponentit tarpeidensa mukaisesti, jotka kootaan valmiiksi tietokoneeksi yrityksen toimesta (Spooner, 1998). Autoteollisuudessa konfiguroitavuus on kasvattanut suosiotaan viime vuosikymmeninä. Asiakkaat voivat valita lukuisista eri moottori, voimalinja, väri ja sisustusvaihtoehdoista parhaiten heidän mieltymyksiinsä sopivan. Wolkswagen hyödyntää modulaarisuutta käyttämällä samaa peruspohjaa mooneen eri automalliin, joiden muut komponentit konfiguroidaan mallin ja tilauksen mukaan (Winterkorn & Pötsch, 2012).

Ikea hyödyntää modulaarisuutta huonekaluissa tarjotakseen kilpailijoita edullisempia ja personoitavampia tuotteita. Huonekalut koostuvat samasta rungosta, johon voi valita haluamiaan lisäosia, kuten ovia, välitasoja, pintakäsittelyjä. Ikea ei kuitenkaan hyödynnä Cto toimintatapaa valmistuksessa vaan ulkoistaa sen asiakkaalle.

3. MODULAARISEN TUOTERAKENTEEN SUUNNITTELU

Modulaarisen tuotteen suunnittelemiseen on esitelty monia erilaisia keinoja. Jose & Tollanaere (2005) jakoivat suunnittelulähestymistapojen tyypit klusterointi-, kuvaaja- ja matriisipohjaisiin, matemaattisiin, tekoälypohjaisiin, geneettisiin algoritmeihin ja muihin lähestymistapoihin. Pakkanen et al. (2016) ehdottavat modularisoinnin lähestymistapojen jakamista neljään kategoriaan: funktiokeskeisiin, indeksipohjaisiin, matriisi ja optimointipohjaisiin lähestymistapoihin. Tässä työssä keskitytään tähän jakoon sen yksinkertaisuuden takia. Käydään läpi myös Juuti et al. (2019) esittämä seitsemän periaatetta modulaaristen rakenteiden suunnitteluun ja asiakas-keskeinen tuotekehitys asiakastyytyvyyden takaamiseksi.

3.1 7 Periaatetta modulaaristen rakenteiden suunnitteluun

Juuti et al. (2019) kokosivat havaintonsa hyvistä käytännöistä modulaarisesta tuotteen suunnittelussa seitsemäksi periaatteeksi. Periaatteet esitellään tässä työssä, koska asiaa tutkittuani päädyin samaan lopputulokseen. Periaatteita noudattamalla saavutetaan yleisesti pätevä tuotesuunnitelma modulaariselle tuoterakenteelle.

Kehitä moduuleja, jotka voivat suorittaa kokonaisen funktion. Tuotekonsepti jaetaan heidän mukaansa funktioihin, jotka taas jaetaan alifunktioihin (Habib, Komoto, 2014). Jakamista pienempiin osiin jatketaan niin kauan, kunnes yhdestä funktiosta vastaa yksi osa. Tässä vaiheessa on saatu piirrettyä kartta funktioista ja niitä suorittavien komponenttien suhteista. Funktiot jaetaan kahteen kategoriaan: pää- ja apufunktioihin (Kamrani, Nasr, 2010). Pääfunktiot ovat funktioita, jotka suo- raan tukevat tai mahdollistavat konseptin funktionaalista tavoitetta. Apufunktiot tu- kevat konseptin funktionaalista tavoitetta epäsuorasti tai vaihtoehtoisesti ne tukevat pääfunktoita (Kamrani, Nasr, 2010).

Funktionaalisen kartan muodostamisen jälkeen on helppo nähdä minkälaisia tehtä- viä pitää suorittaa ja jakaa tehtävät moduulien suoritettavaksi. Periaatteen mukaan toimimalla varmistutaan siitä, että kaikilla moduuleilla on tehtävä ja saadaan määri- tettyä niille selvät rajapinnat.

Kehitä pohja tuotteelle, johon muut moduulit kiinnittyvät. Erixonin (1998) mukaan moduulit voidaan luokitella joko pohjakokoonpanoon tai toisiinsa liittyvään kokoonpanoon. Toisiinsa liittyvässä kokoonpanossa moduulit kiinnittyvät toisiinsa pohjan sijasta. Tällä tekniikalla valmistettaessa variaatiolla voi olla odottamattomia seurauksia ja moduulien välisten rajapintojen hallinta muodostuu monimutkaiseksi. Tämän tilanteen korjaamaan on pohjakokoonpano, jonka etuja ovat nimenomaan rajapintojen ja variaation vaikutusten hallinta. Moduulit voidaan suunnitella sopimaan vain pohjaan, jolloin niiden keskinäisiä suhteita ei tarvitse miettiä niin paljon. (Juuti et al., 2019)

Suunnittele päämoduulit keskenään vaihdettaviksi. Asiakkaat odottavat modulaarisesta vaihtoehdosta kuullessa voivansa konfiguroida tuotteensa hyvinkin laajasti (Lindh, 1992). Juuti et al. (2019) käyttää esimerkkinä autoteollisuutta, jossa asiakkaat voivat haluta erilaisia hyttejä, moottoreita ja voimasiirron vaihtoehtoja. Heidän mukaansa tehokkaan kustomoinnin saavuttamiseksi päämoduulien rajapinnat on oltava keskenään samankaltaisia, jotta ne voidaan kiinnittää pohjamoduuliin eli autoteollisuuden tapauksessa auton runkoon.

Varaa tilaa tuotteessa erilaisille moduuleille. Jotta kaikki suunnitellut konfiguraatiot saadaan sijoitettua paikoilleen, Förg et al. (2014) ehdottaa layout standardisointia tuotteen sisällä. Sen tarkoituksena on kartoittaa, mihin ja missä asennossa mikäkin komponentti asetellaan, jotta voidaan varmistaa layouteille riittävästi tilaa. Layout standardisoinnissa kartoitetaan kaikki layout vaihtoehdot ja niiden ominaisuudet sekä huomioidaan niiden myyntivolyymit ja varianttien priorisointi (Förg et al., 2014). Förg et al. (2014) Toteavat joustamattoman layoutin muodostuksen eli standardisoinnin olevan vaihtoehtoista kustannustehokkain.

Määritä alueet, joille muutoksia ei voi tehdä. Holmqvist (2004) esitti Juuti et al. (2019) mukaan idean tuotteen jakamisesta lohkoihin, jossa yksi lohko vastaisi tietynlaisesta variaatiosta. Lohkojen muodostamisen ideana on, ettei yhden lohkon variaatio vaikuta muihin lohkoihin, jotta vaihtoehtoiset moduulit voidaan kiinnittää paikoilleen muutosten jälkeenkin (Juuti et al., 2019). Moduuleille on siis asetettava tietyt rajapinnat, joita ei voida muuttaa sekä kokoluokka, jonka sisälle sen on mahdollista.

Hyödynnä standardisointi alueilla, joilla asiakas ei odota variaatiota. Organisaation sisäisen variaation, eli kuinka paljon erilaisia komponentteja käytetään tuotteen kokoamiseen vaikuttaa suuresti sen kustannuksiin (Winterkorn & Pötsch,

2012). Kustannuksia voidaan pienentää vähentämällä sisäistä variaatiota kokoonpanossa ja saavuttaa näin mittakaavaetua kilpailijoihin (Winterkorn & Pötsch, 2012). Standardoidut osat mahdollistavat kustannuksien pienentämistä ja tästä syystä niitä tulisi käyttää alueilla, joilla asiakas ei osaa odottaa variaatiota eikä tarvitse sitä.

Suunnittele standardoidut paikat ja liitokset lisävarusteille, johdoille ja letkuille. Juuti et al. (2019) mukaan (United States Department of Defense, 1995) standardoitujen sijaintien käyttäminen mahdollistaa tuotannon aloittamisen, vaikkei vielä tiedettäisi, mitä lisävarusteita pitäisi olla saatavilla. Se tukee myös tuotannossa kokoonpanon standardisointia (Juuti et al., 2019).

1.	Kehitä moduuleja, jotka voivat suorittaa kokonaisen funktion	(Pahl & Beitz, 2013) (Ulrich & Eppinger, 2008)
2.	Kehitä pohja tuotteelle, johon muut moduulit kiinnittyvät	(Erixon, 1998)
3.	Suunnittele päämoduulit keskenään vaihdettaviksi	(Lindh, 1992)
4.	Varaa tilaa tuotteessa erilaisille moduuleille	(Förg et al., 2014)
5.	Määritä alueet, joille muutoksia ei voi tehdä	(Holmqvist, 2004)
6.	Hyödynnä standardisointi alueilla, joilla asiakas ei odota variaatiota	(Winterkorn and Pötsch, 2012)

7.	Suunnittele standardeidut paikat ja liitokset lisävarusteille, johdoille ja letkuille	(United States Department of Defense, 1995)
----	---	---

Taulukko 1: 7 periaatetta moduulien suunnitteluun (Juuti et al., 2016)

3.2 Asiakaslähtöinen tuotekehitys

Asiakaslähtöisen tuotekehityksen hyödyntäminen on tärkeää asiakastyytyväisyyden takaamiseksi. Asiakaslähtöisen tuotekehityksen ajatuksen ydin on luoda sellaisia tuotteita, jotka vastaisivat asiakkaan tarpeita mahdollisimman hyvin. Tämän filosofian keskiössä on termi nimeltään asiakkaan ääni, jonka tehtävänä on muuntaa asiakkaan tarpeet parametreiksi, joiden täyttymistä voidaan mitata ja seurata (Chaudha et al., 2011). Asiakaslähtöisen tuotekehityksen keskuksena toimii laadun talo, joka kerää asiakastarpeet, niitä vastaavat tuoteominaisuudet, vertaa niitä kilpailijoiden tuotteisiin ja antaa tulokseksi ominaisuuksien tärkeysjärjestyksen (Ghaudha et al., 2011).

Garverin (2012) mukaan asiakaslähtöinen tuotekehitys vastaa yleisesti kysymyksiin: asiakkaan tarpeista (mitä), vaatimuksien merkityksestä asiakkaalle (miksi) ja tuotteen ominaisuuksista, joita tarvitaan asiakkaan tarpeiden tyydyttämiseen (miten). Laatufunktion keskuksena toimii laadun talo (HOQ). HOQ keskittyy keräämään, ymmärtämään ja käyttämään asiakkaan tahtoa suunnitteluprosessissa (Garver, 2012). HOQ sisältää tärkeät kysymykset asiakastyytyväisyyden takaamiseksi ja selvittää niiden välisiä suhteita seitsemän osaiseen matriisiin. Seitsemää elementtiä käyttämällä HOQ:sta tulee informaation arkisto, jota voidaan käyttää mekanismina suunnittelussa (Kamrani & Nasr, 2010).

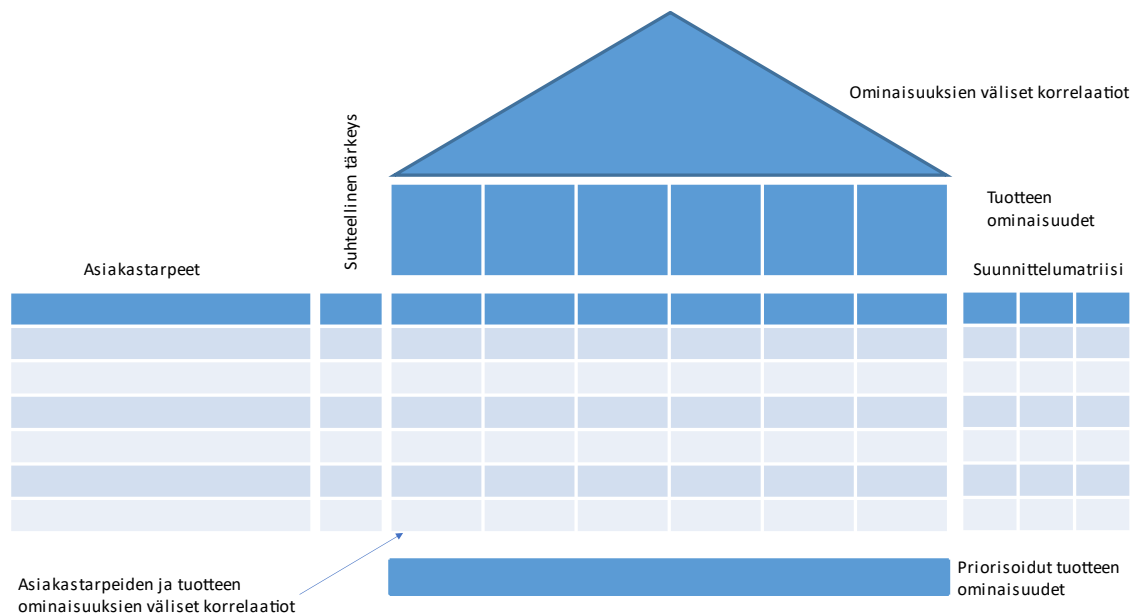
Asiakastarpeet ja niiden arviointi

Asiakastarpeet muodostuvat niistä asioista, joita asiakas toivoo tuotteelta. Ne voivat liittyä esimerkiksi suorituskykyyn, hintaan tai painoon. Asiakastarpeet ne järjestellään matriisiin. Asiakastarpeet eivät ole kaikki asiakkaalle yhtä tärkeitä ja onkin tärkeä saada tietää, mitkä asiat ovat asiakkaalle kaikkein tärkeimpiä, jotta oikeat ominaisuudet voidaan priorisoida (Garver, 2012). (Garver, 2012) kertoo yleisimmän tavan olevan kysely, jossa asiakas kertoo, mitkä tarpeista ovat tärkeitä ja mitkä eivät niin tärkeitä. Se kuitenkin aiheuttaa hänen mukaansa priorisointiongel-

mia, sillä asiakkaat arvottavat usein kaikki tarpeensa hyvin tärkeiksi. Tämän heikkouden takia Garver (2012) ehdottaa asiakastarpeiden suhteellisen tärkeyden määrittämisessä käytettävän Maximum Difference Scaling tapaa, jossa asiakkaan tarpeet ryhmitetään viiden tarpeen ryhmiin ja pyydetään valitsemaan niistä tärkein ja vähiten tärkeä (Garver, 2012). Tällä tavalla saadaan tehokkaasti tuotettua tulokset, jolla priorisointi voidaan suorittaa.

Tuotteen ominaisuudet ja korrelaatiomatriisi

Tuotteen ominaisuudet ovat toiselta nimeltään suunnitteluvaatimuksia ja niitä käytetään tutkimaan suunnittelun onnistumista vertaamalla tuotteen ominaisuuksia asiakkaan tarpeisiin (Kamrani & Nasr, 2010). Tuotteen ominaisuuksien on oltava mitattavaan muotoon merkittynä, jotta vertaus voidaan suorittaa. Kuvassa 2 esitetään laadun talon sijoittuvat elementit, jotka tässä luvussa esitellään. Tuotteen ominaisuudet sijaitsevat laadun talossa katon alla. Tuotteiden ominaisuuksien vaikutuksia toisiinsa kuvataan korrelaatiomatriisilla, joka on kolmiomainen matriisi ja toimii laadun talon kattona. Ominaisuuksien välinen korrelaatio eli toiseen ominaisuuteen tehdyn muutoksen vaikutukset toiseen ominaisuuteen ratkaistaan korrelaatiomatriisilla (Franceschini et al., 2015). Tuotteen ominaisuuksien vaikutusta toisiinsa kuvataan symboleilla. Matriisi paljastaa, mitkä ominaisuudet tukevat toisiaan ja mitkä ovat konfliktissa (Franceschini et al., 2015). Konfliktissa olevat ominaisuudet voivat johtua asiakastarpeiden ristiriidoista, jolloin kompromisseja joudutaan tekemään.



Kuva 2: Laadun talo (Garver, 2012)

Suunnitelu- ja suhdematriisi

Suunnittelumatriisissa verrataan kilpailijan tuotetta ja verrataan sitä asiakkaan tarpeisiin. Näin saadaan selville, kuinka paljon täytyy milläkin alueella parantaa.

Laadun talon rakentaminen vaatii asiakkaan tarpeiden ja tuotteen ominaisuuksien välisten suhteiden selvittämisen suhdematriisilla (Franceschini et al., 2015). Asiakasvaatimukset ja tuotteen ominaisuudet verrataan toisiinsa ja niiden väliset suhteet määritetään. Yleensä käytetään symboleja kuvaamaan suhteen voimakkuutta. Suhteiden voimakkuudet jaetaan voimakkaaseen keskivahvaan ja heikkoon. Seuraavaksi symbolit korvataan numeroilla. (Franceschini et al., 2015)

Valmistunutta suhdematriisia voidaan tarkastella tyhjien sarakkeiden tai rivien varalta. Tyhjä rivi tarkoittaa, ettei yksikään tuotteen ominaisuus täytä asiakkaan tarvetta. Tyhjä sarake taas tarkoittaa, ettei tuotteen ominaisuus vastaa yhteenkään asiakkaan tarpeeseen ja se voidaan poistaa turhana laadun talosta. (Kamrani & Nasr, 2010)

Kilpailijoiden arviointi

Kilpailijoiden arviointi jakaantuu kahteen osaan: tekniseen benchmarkingiin ja asiakaslähtöiseen kilpailija-arviointiin (Franceschini et al., 2015). Teknisessä benchmarkingiin kehitettävää tuotetta ja sen ominaisuuksia verrataan suorituskyyvyltään kilpailijaan (Franceschini et al., 2015). Näiden tietojen pohjalta asetetaan tavoitearvot, joita käytetään syötteenä lopulliseen suunnitelmaan.

Asiakaslähtöisessä kilpailijavertailussa on tiivistelmä kovimpien kilpailijoiden tuotteiden ominaisuuksista suhteessa kehitettävään tuotteeseen (Kamrani, Nasr, 2010). Kilpailijavertailu sisältää myös kilpailijoiden ja designin vertailun asiakasvaatimusten täyttämisen suhteen (Kamrani & Nasr, 2010).

Priorisoidut ominaisuudet

Kaikki edelliset asiat otetaan huomioon kohdassa priorisoidut ominaisuudet. Se sisältää toteutuksen vaikeuden arvioinnin, tärkeyden arvioinnin ja kustannusarvion (Ginting et al., 2019). Tulokset sisällytetään matriisiin numeroina kuvaamaan tiettyä prosenttiväliä (Ginting et al., 2019). Kamrani ja Nasr (2010) ratkaisevat priorisoitavat ominaisuudet absoluuttisen painon kaavalla:

$$a_j = \sum_{i=1}^n R_{ij} C_i, \quad (1)$$

jossa a_j on absoluuttinen paino tuoteominaisuudelle ($j=1, \dots, m$), R_{ij} on paino suhdematriisissa ($i=1, \dots, n, j=1, \dots, m$), C_i on asiakasvaatimuksen tärkeys ($i=1, \dots, n$),

m on tuoteominaisuuksien määrä ja n on asiakasvaatimusten määrä. (Kamrani & Nasr, 2010)

3.3 Funktiokeskeiset työkalut

Funktiokeskeiset modularisointi työkalut keskittyvät jakamaan tuotekonseptin funktioihin ja useisiin alifunktioihin (Habib, Komoto, 2014). Niiden tarkoituksena on saada yksi funktio vastaamaan yhdestä tuotteen ominaisuudesta, jotta vältetään funktioiden päällekkäisyydeltä ja täten kompleksisuuden kasvamiselta. Tyypillistä on selvittää yhteyksiä visualisoida funktioiden, alifunktioiden ja komponenttien suhteita suhdekartoilla. Suunnittelutehtävien purkaminen funktionaalisesti etenee seuraavasti. Ensimmäisessä vaiheessa on esittää se yhtenä laatikkona, jolle kuvataan tämän funktion sisääntulot ja ulostulot, olivat ne energiaa, materiaalia tai signaaleja (Ulrich & Eppinger, 2012). Energiaa, materiaaleja ja signaaleja ajatellaan heidän mukaansa funktionaalisissa työkaluissa virtana, joka virtaa tuotteen läpi ja tässä luvussa niistä käytetään yleistä nimitystä virta. Seuraavaksi tämä yksittäinen kokonaisfunktio useampaan kokonaisfunktioon, jotta saadaan tarkempi kuvaus tuotteen toiminnasta toteuttaakseen kokonaisfunktion (Ulrich & Eppinger, 2012). Jokainen alifunktio voidaan yleensä jakaa vielä yksinkertaisempiin alifunktioihin. Jako prosessia jatketaan, kunnes jokainen alifunktio on riittävän yksinkertainen käsiteltäväksi. Optimaalinen määrä alifunktioita diagrammissa on 3–10 (Ulrich & Eppinger, 2012). Tässä luvussa esitellään kolme erilaista tuoterakenteen kehitysmallia. Ulrich & Eppingerin (2012) ja Stone et al. (2000) malleja vertaillaan ensin keskenään, sillä ne ovat melko samanlaisia. Viimeiseksi esitellään Blees (2010) kehittämä PKT malli erikseen sen suurempien eroavuuksien takia.

Ulrich & Eppinger (2012) vs. Stone et al. (2000)

Kaikki kolme esitettävää mallia (Ulrich & Eppinger, 2012), Blees (2010) & Kraus (2011), (Stone et al., 2000) lähtevät tuoterakenteen kehityksessä liikkeelle tuoterakenteen purkamisesta funktioihin ja alifunktioihin. Prosessi etenee, kuten aiemmin mainittiin.

Toisin kuin Ulrich & Eppinger (2012) funktioketjut järjestetään ajan mukaan. Yleisesti, kuten Pahl & Beitz (2013) lähestymistavassa, seurataan virtoja alifunktioiden

läpi huomioimatta alifunktioiden riippuvuutta teitystä järjestyksestä. Ulrich & Eppinger tuotekehitysprosessit ovat joko rinnakkaisia, peräkkäisiä tai yhdistettyjä. Stone et al. (2000) laajentavat tätä konseptia alifunktioihin ja virtoihin funktionaalisessa mallissa. Lopuksi funktiot kokoamisen yhteen funktionaaliseksi malliksi. Tässä vaiheessa saattaa vaatia uusien alifunktioiden lisäämistä.

Toisessa vaiheessa Ulrich & Eppinger (2012) keskittyvät kehittämään ratkaisuideoita. Niitä etsitään sekä yrityksen sisäisesti heidän kuvaamissaan aivoriihi sessioissa, jossa yrityksen työntekijät pyrkivät luomaan mahdollisimman paljon erilaisia ratkaisuvaihtoehtoja. Ulkoisessa ideoiden etsinnässä vertaillaan kilpailijoiden tuotteita, tutkitaan patenteja ja haastatellaan sekä asiantuntijoita että asiantuntevia asiakkaita.

Stone et al. (2000) ei tarjoa työkaluja ratkaisuvaihtoehtojen kehittämiseen vaan keskittyvät tarjoamaan kolme heuristiikkaa tuoterakenteen tunnistamiseen.

Kolmannessa vaiheessa (Ulrich & Eppinger, 2012) mallissa tutkitaan systemaattisesti mahdolliset ratkaisut. Tähän käytetään kahta työkalua: konseptin luokittelu puu ja konseptin yhdistely taulukko. Puuta käytetään jakamaan kaikki mahdolliset ratkaisut muutamaan kategoriaan, joita voidaan vertailla ja karsia. Taulukko ohjaa suunnittelijoita valikoivasti harkitsemaan eri alifunktioiden yhdistelmiä. Se sisältää toisiinsa alifunktion ratkaisuehdotuksia allekkain sarakkeessa. Tähän alifunktioon liittyvät alifunktiot asetetaan vierekkäisille sarakkeille. Nyt näiden ratkaisujen välille voidaan vetää erilaisia kombinaatioita kokonaisfunktion saavuttamiseksi (Ulrich & Eppinger, 2012).

Stone et al. (2000) kolme heuristiikkaa moduulien määrittämiseksi:

Heuristiikka 1: Moduuli voidaan määrittää seuraamalla virtaa, joka tulee systeemiin. Ne alifunktiot, joiden läpi virta kulkee, kunnes se muuntuu tai poistuu systeemistä muodostavat moduulin.

Heuristiikka 2: Rinnakkaisen funktioketjun haarat muodostavat moduuleja. Rajapinta moduulille muodostuu täten haarautumispisteessä.

Heuristiikka 3: Tunnista alifunktiot, jotka muuntavat virtaa. Jos sellaisia ei löydy, löydetty muuntava alifunktio on moduuli itsessään.

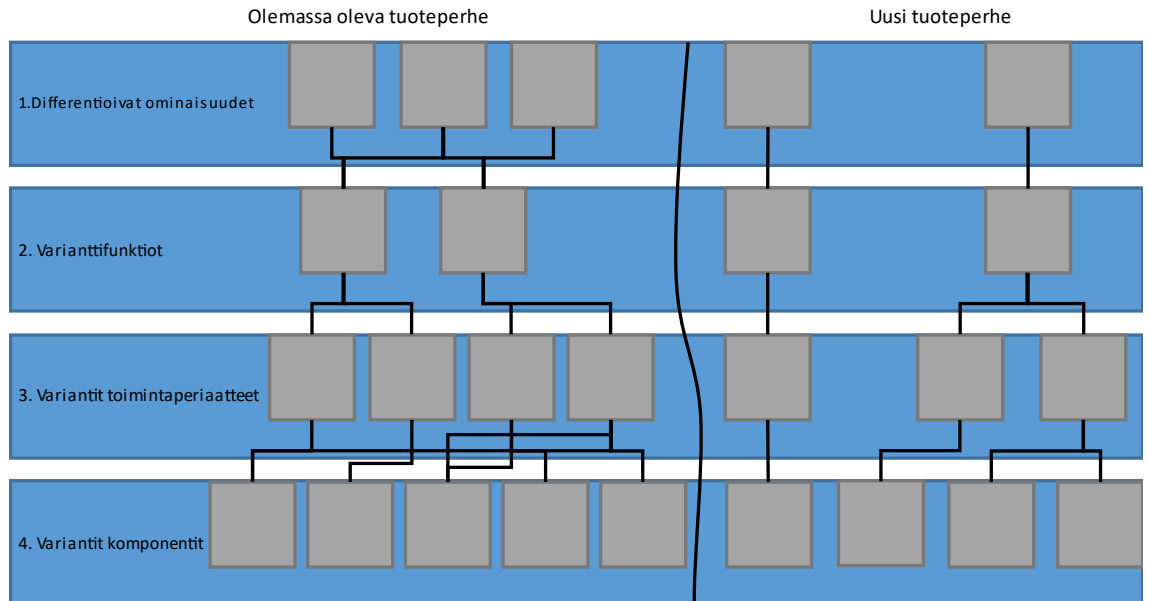
PKT-malli

Pkt-malli on Bleesin et al. (2010), Krausen ja Eilmuksen (2011) kehittämä modularisointi työkalu, joka toteuttaa modularisoinnin funktioiden näkökulmasta. Pkt-malli

jakautuu kahteen vaiheeseen: ensimmäisessä vaiheessa pyritään vähentämään sisäistä variaatiota purkamalla se funktioihin ja niitä toteuttaviin elementteihin, jonka jälkeen tuoteperhe suunnitellaan uudelleen vastaamaan paremmin optimaalista variaatiota (Eilmus et al., 2012).

Eilmuksen et al., (2012) mukaan prosessi lähtee liikkeelle olemassa olevan tuotteen analysoinnista. Luodaan variaatiopuu, joka esittää ulkoisen variaation ja yhdistää asiakkaan kokemat differentioivat tuotteen ominaisuudet ja erilaiset tuotevariaatiot. Kun ulkoinen variaatio on selvillä, siirrytään tutkimaan tuotteen sisäistä variaatiota. Se jaetaan kolmelle tasolle: funktioihin, toimintaperiaatteisiin ja komponentteihin. Funktiot johdetaan käyttämällä tuoteperheen funktiorakennetta, joka esittää jokaisen funktion variaation. Selvitetyt yhteydet esitetään moduulien rajapintakaavioilla. (Eilmus et al., 2012)

Eilmuksen et al. (2012) mukaan tämän jälkeen differentioituvat ominaisuudet, funktiot, toimintaperiaatteet ja komponentit sijoitetaan variaation allokointimalliin, jossa niiden väliset yhteydet visualisoidaan ja analysoidaan. Kuvassa 3 on esitettyä esimerkkitapaus siitä, miltä uusi tuoteperheen rakenne voi näyttää, kun käytetään variaation allokointimallia rakenteen uudelleenjärjestelyssä. Löydetyt heikkoudet voidaan ratkaista suunnittelemalla niihin kytkeytyvät osat uudestaan (Eilmus et al., 2012). Variaation allokointimallin hyödyntäminen ongelmanratkaisussa auttaa variaatiosuuntautuneen tuoteperheen konseptin löytämisessä. Optimaalinen variaatio orientoitunut tuoterakenne saavutetaan, kun yksi funktio vastaa yhdestä tuoteominaisuudesta. (Eilmus et al., 2012)



Kuva 3: Variaation allokontimalli olemassa olevasta ja uudesta tuoteperheestä (Eilmus et al. 2012)

Toisen vaiheen tarkoituksena on kehittää itsenäisesti tuotteen eri elinkaaren vaiheisiin ideaaliset modularisointi konseptit, jotka täyttävät yrityksen tavoitteet juuri siinä elinkaaren vaiheessa (Blees et al., 2010). Konseptit eri elinkaaren vaiheissa voivat erota toisistaan, sillä tällä tavalla vaiheiden yksittäiset tarpeet saadaan täytettyä. Modulaariset konseptit kerätään moduulien prosessikaavioon, jossa ne esitetään elinkaaren vaiheittain. Konseptit pyritään yhdistämään yhdeksi valmiiksi kokonaisuudeksi, mutta usein konseptit ovat ristiriidassa keskenään. Moduulien prosessikarttaa voidaan hyödyntää myös ongelmanratkaisussa, jotta löydettäisiin ratkaisu, joka täyttää riittävästi kaikkien vaiheiden tarpeita. (Blees et al., 2010)

Yhdistävät piirteet

Vaikka PKT malli eroaa melko suuresti sen hyödyntämien työkalujen takia, malleja yhdistää juuri tapa jakaa tuote funktioittain pienempiin osiin ja pyrkiä etsimään mahdollisia moduuleja tätä kautta. Koska niiden perusajatus on sama, niiden heikkoudet ovat myös yhteisiä. Funktionaaliset mallit toimivat hyvin pienemmille tuotteille, koska alifunktioiden määrä pysyy hallittavassa määrässä, kuten Ulrich & Eppinger (2012) totesivat, alle kymmenessä alifunktiossa. Tuotteiden monimutkaisuuden kasvaessa alifunktioiden määrä kasvaa sellaiseksi, ettei työkaluista ole enää hyötyä. Eilmus et al. (2012) tutkimuksessa PKT- mallista ongelmia kohdattiin yli 40 komponenttia sisältävissä tuotteissa. Myöskin Ulrich & Eppinger (2012) toteavat

ettei heidän menetelmällään ole mahdollista käydä läpi kaikkia yhdistelmiä, jotka konseptien yhdistelytaulukosta saadaan.

3.4 Optimointipohjaiset työkalut

Optimointipohjainen tapa luokitella komponentit moduuleihin perustuu ajatukselle, että komponentit, joilla on suuri riippuvuus toisistaan, ryhmitellään omaan moduuliinsa (Kamrani, Nasr, 2010). Optimointipohjaisten lähestymistapojen tarkoituksena on usein algoritmien avustuksella luoda joukko mahdollisia ratkaisuja tuotteelle. Mahdollisten ratkaisujen generoinnin jälkeen ratkaisuja mitataan jollakin ennalta määritetyllä mittarilla. Mittari valitsee parhaat ratkaisut joukosta, jotka siirretään jatkokokehitettäväksi. Loput poistetaan. Tällaisia iteraatiokierroksia jatketaan, kunnes saavutetaan optimaalinen ratkaisu tai kunnes resurssit loppuvat ja valitaan siihen mennessä paras mahdollinen ratkaisu. Tässä kappaleessa esitetään Gonzalez-Zugastin et al. (2000) kehittämä optimointimalli tuoteperheiden suunnitteluun, yksityiskohtaisempi Pareto-optimointi malli ja verrataan sitä nykyaikaisempaan geneettiseen algoritmiin pohjautuvaan optimointiin.

Gonzalez-Zugastin et al. (2000) interaktiivinen tuotesuunnittelu sisältää neljä vaihetta: 1. Vaatimusten ja mallin keräys 2. Tuotealustan suunnittelu 3. Varianttien suunnittelu 4. Tuotealustan arviointi ja uudelleen neuvottelu.

Ensimmäisessä vaiheessa koostetaan matemaattiset mallit systeemeille, jotka yhdistävät suunnittelussa tehdyt valinnat suorituskykyindekseiksi tuotteille (Gonzalez-Zugasti et al., 2000). Mallit sisältävät suorituskyvyn, kulut ja kaupallisissa tuotteissa sisältäisi myös liikevaihdon ja kilpailijat. Näillä malleilla ja vaatimuksilla jokaiselle tuotteelle, voidaan luoda ryhmä yksittäin suunniteltuja tuotteita, jota vastaan tuoteperheen varianttia verrataan. (Gonzalez-Zugasti et al., 2000)

Toisessa vaiheessa määritetään tuoteperheen alusta samoilla vaatimuksilla kuin ensimmäisen vaiheen yksittäiset tuotteet. Suunnittelijat ja erilaiset asiantuntijat päättävät, mitkä osat suunnitelmasta muodostuvat yhteiseksi tuoteperheen alustaksi. Päätöksiä tehdessä on otettava huomioon vaatimusten samankaltaisuus, alijärjestelmien joustavuus ja muita asioita kuten resurssien saatavuus ja aikataulun aiheuttamat haasteet. (Gonzalez-Zugasti et al., 2000)

Kolmannessa vaiheessa lukitaan jo määritelty tuoteperheen alusta, eikä sille tehdä enää muutoksia. Alusta annetaan suunnittelutiimeille, jotka voivat nyt viimeistellä ja optimoida suunnitelmat omasta osiostaan tuoteperheessä. (Gonzalez-Zugasti et al., 2000)

Neljännessä vaiheessa Uudet suunnitelmat muodostavat uuden vaihtoehtoisen tuoteperheen (Gonzalez-Zugasti et al., 2000). Uutta tuoteperhettä verrataan aikaisemmin suunniteltuihin yksittäisiin tuotteisiin suorituskyvyn, kulujen, riskien ja muiden asioiden osalta. Jos uusi tuoteperhe ei saavuta haluttua taso vertailussa, voidaan palata askelissa taaksepäin ja harkita uudestaan päätöksiä, joita suunnitellaessa tuotetta on tehty. Tällaisia iterointikierroksia voidaan suorittaa niin kauan, kunnes saavutaan hyvään eli **optimoituun lopputulokseen**. (Gonzalez-Zugasti et al., 2000)

Pareto-optimaalinen lähestymistapa

Gonzalez-Zugasti et al. (2000) esittämä malli pohjautuu suunnittelutiimien tekemään iterointiin. Nykyaikana kuitenkin on mahdollista löytää nämä optimaaliset lopputulokset nopeammin hyödyntämällä tietokoneita iteroinnissa. Rai ja Allada (2003) ehdottivat optimointialgoritmien käyttöä tuotekehityksessä. Ytimeltään tämä oliopohjainen algoritmi pyrkii levittämään Pareto-optimaalia iteratiivisen prosessin kautta samalla säilyttäen ratkaisupopulaation monimuotoisuuden (Rai & Allada, 2003). Pareto-optimaalinen ratkaisu tarkoittaa sellaista tilannetta, jossa ratkaisun yhtäkään osa-aluetta ei voida parantaa huonontamalla toista.

Prosessi alkaa, kun luoja olio luo joukon tuotevaihtoehtoja sille asetettujen rajoitteiden puitteissa (Rai & Allada, 2003). Tuotevaihtoehdot kuvataan moduuleista koostuvaksi vektoriksi. Seuraavaksi ratkaisut tarkastetaan, löytyvätkö kaikki tuotteen tarvittavat toiminnot tältä tuotevaihtoehdolta. Tuotevaihtoehto läpäisee tarkistuksen, jos siltä löytyy tarvittava toiminnallisuus eikä kopiota siitä ole jo olemassa. Muutoin se lähetetään takaisin muokattavaksi. (Rai & Allada, 2003)

Seuraavassa vaiheessa tuotevaihtoehdot arvioidaan asiakkaan asettamien tavoitteiden/mittareiden perusteella (Rai & Allada, 2003). Tuotevaihtoehdot jaetaan nyt Pareto-optimaalisiin ja ei Pareto-optimaalisiin. Pareto-optimaalinen vaihtoehto tarkoittaa sellaista vaihtoehtoa, joka ei ole verrokiaan huonompi missään tehtävässä ja on sitä parempi ainakin yhdessä koko tutkinta-alueella. Pareto-optimaaliset vaihtoehdot/ratkaisut säilytetään seuraavia iteraatiokierroksia varten. Ei pareto-optimaaliset ratkaisut/vaihtoehdot lähetetään muokattavaksi oliolle, joka muokkaa niitä sille asetetun strategian mukaisesti. Nämä muokatut ratkaisut/vaihtoehdot jaetaan seuraavaksi hyviin ja huonoihin. Huonot poistetaan järjestelmästä ja hyvät paloitellaan ja niistä poistetaan huonot/verrokkejaan huonommat moduulit. (Rai & Allada, 2003)

Jäljelle jääneet osittaiset design ratkaisut lähetetään takaisin luoja oliolle, joka täyttää tyhjät osat uusilla moduuleilla. Iteroiva prosessi jatkuu, kunnes systeemi supenee kohti ratkaisua. (Rai & Allada, 2003)

Geneettiset algoritmit

Geneettiset algoritmit ovat viime vuosikymmenenä suosiota kerännyt tapa etsiä moduuleja. Geneettisiä algoritmeja hyödynnetään monesti Design structure matriisien kanssa (Cheng et al., 2012; Binay et al., 2018). Ne ovat luonteeltaan hyvin samankaltaisia perinteisten optimointialgoritmien kanssa. Nekin arvioivat ratkaisun/vaihtoehdon hyvyttä tietyillä sille asetetuilla mittareilla. Geneettisillä algoritmeilla tätä kutsutaan "fitness funktioksi". Evolutionääriset algoritmit pyrkivät ottamaan huomioon suuremman määrän mahdollisia ratkaisuja ja täten todennäköisemmin löytävät optimaalisen ratkaisun (Kamrani & Nasr, 2010). Ne pyrkivät nimensä mukaan jäljittelemään evoluutioprosessia. Tämän on mahdollistanut kasvanut tietokoneiden laskentateho ja tekoälyn kehittyminen.

Suurin ero perinteisten optimointialgoritmien ja geneettisten algoritmien välillä on prosessi, jolla algoritmi muokkaa ei optimaalista ratkaisua järjestelmässä. Optimointialgoritmit sisältävät olion, joka suorittaa sille asetetun strategian mukaan muutoksia ratkaisulle, kun taas geneettiset algoritmit valitsevat ratkaisusta/vaihtoehdosta satunnaisesti geenejä, eli komponentteja (Binay et al., 2018). Satunnaisesti valittujen geenien avulla luodaan vaihtoehtoisia tuoteratkaisuja generoimalla kaikki mahdolliset permutaatiot valituista geeneistä. Geenien paikkoja siis vaihdetaan keskenään ja arvioidaan sitten, mikä näistä vaihtoehdoista on asetettujen mittareiden mukaan paras jälkeläinen. Paras jälkeläinen siirtyy seuraavalle iteraatiokierrökselle. Tätä iteraatioprosessia jatketaan, kunnes ratkaisut suppenevat kohti tiettyä tuoteratkaisua. (Binay et al., 2018)

3.5 Matriisipohjaiset työkalut

Matriisipohjaiset työkalut sisältävät nimensä mukaisesti matriisin, jonka avulla tuotteen osien ja osakokoonpanojen välisiä suhteita selvitetään moduulien määrittämiseksi. Stewardin (1981) esittämä Design Structure Matriisin (DSM) hyödyntämisestä systeemien suunnittelussa on muodostunut yleisimmäksi keinoksi modularisointiin matriiseja hyödyntäen. DSM hyödyntää useampaa matriisia: ensimmäi-

nen kartoittaa muuttujien välisiä suhteita ja toista käytetään Stewardsin (1981) mallissa peruselementtien eli moduulien löytämiseen. Modernit DSM lähestymistavat voivat käyttää vielä useampia matriiseja, kuten alla esitettyssä Cheng et al. (2012) mallissa.

Pakkasen et al. (2016) mukaan matriisit ovat hyödyllisiä suhteiden kartoittamiseen ja rajapintojen löytämiseen moduulien välillä. Muuttujien määrän kasvaessa, matriiseista tulee kuitenkin vaikeita lukea ja käyttää (Pakkanen et al., 2016). Tuoterakenteen päätöksissä käytetään algoritmeja, jotka järjestelevät matriiseja ja osittelevat ne helpottaen näin moduulien löytämistä.

Pakkanen et al. (2016) toteavat matriisipohjaisten keinojen olevan riittämättömiä kokonaisvaltaisen modularisointiin, koska ne eivät ota huomioon, kuinka kerätään tieto, jota moduulien löytämiseen tarvitaan. DSM lähestymistavat eivät myöskään ota huomioon komponenttien geometrioita, eivätkä täten sovellu varsinaiseen moduulien suunnitteluun. Modulaarinen tuoteperhe tarvitsee tapauskohtaisen tavoitteiden asettamisen, joka selittää, miksi tuoteperhe ositeltaisiin tietyllä tavalla. (Pakkanen et al., 2016). Ne ovat kuitenkin löytävät nopeasti vaihtoehtoisia ratkaisuja, joita suunnittelijoiden johdolla jatkojalostaa tai hyödyntää sellaisenaan.

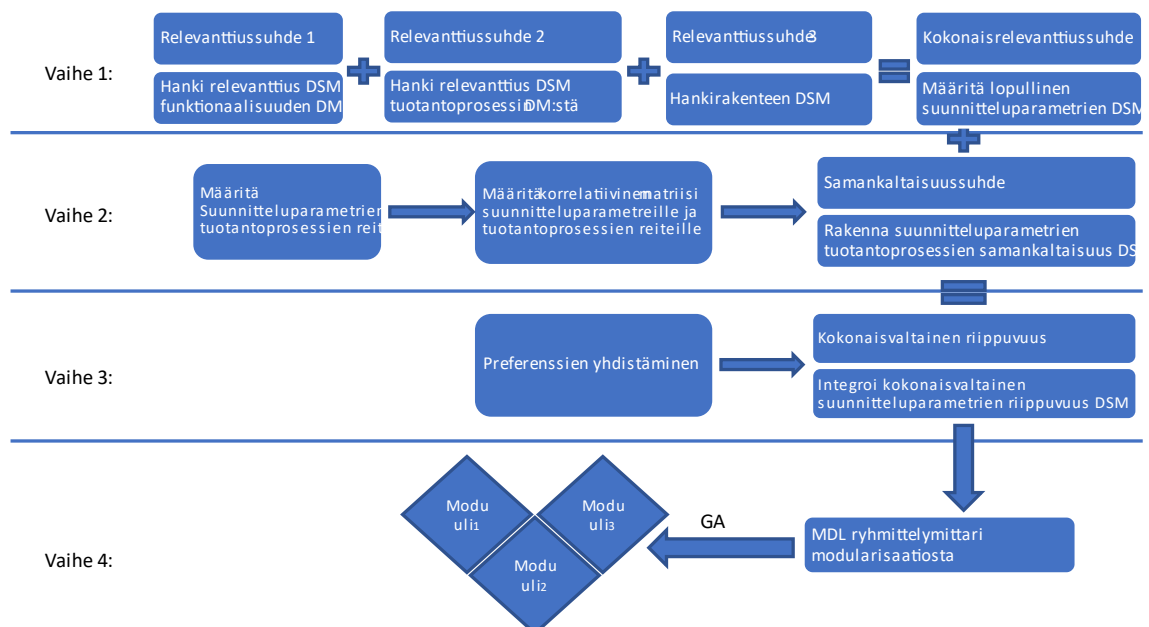
3.5.1 Design structure matrix ratkaisu

Design structure matriisi on tyypillisin tapa hyödyntää matriiseja tuotesuunnittelussa ja tuotteen modularisoinnissa. Se kehitettiin 1970-luvun lopulla ja ensimmäiset sovellukset teollisuudessa syntyivät 1990-luvun alussa MIT tutkijoiden toimesta, jotka hyödynsivät DSM:ää projekteissa auto-, elektroniikka- ja lentokone-teollisuudessa (Eppinger et al. 2012). Nykyään DSM on laajasti käytetty ja maailmanlaajuisesti käytetty työkalu tuotesuunnittelussa, mutta myös monenlaisten muiden monimutkaisten systeemien suunnittelussa (Eppinger et al., 2012). Design structure matriisi koostuu $N \times N$ matriisista, joka kartoittaa vuorovaikutuksia N määrässä systeemin elementtejä (Eppinger et al., 2012). Tuoterakenteen suunnittelun tapauksessa elementtejä ovat tuotteen komponentit ja vuorovaikutuksia ovat rajapinnat osien välissä.

Eppingerin et al. (2012) mukaan DSM-sovellukset voidaan yleisesti jakaa viiteen vaiheeseen: 1) Systeemin purkaminen osasysteemeihin ja osiin 2) Elementtien vuorovaikutussuhteiden tunnistaminen ja dokumentointi 3) elementtien järjestäminen uudestaan siten, että niiden vuorovaikutuksesta voidaan löytää kuvioita ja mal-

leja systeemin toiminnalle 4) hyödyllisen esitelmän luominen DSM mallista korostaen tärkeimpiä ominaisuuksia 5) systeemin parantaminen kerätyn tiedon perusteella.

(Cheng et al., 2012) esittävät aksiomaattiseen suunnittelun teoriaan ja Design structure Matriisin pohjautuvan lähestymistavan moduulien tunnistamiseen. Tätä lähestymistapaa verrataan Williamssonin ja Sellgrenin (2016) kehittämään malliin, joka hyödyntää design structure matriisia ja yhdistää sen aiemmin esiteltyyn laadun talon matriisin kanssa sekä Helmer et al. (2010) lähestymistapaan, jonka erikoisuutena on sen hyödyntämä epäsymmetrinen matriisi. Kuvassa 4 on kuvitettuna Cheng et al. (2012) esittämä lähestymistapa modularisointiin Design structure matriisien avulla. Se jakaantuu neljään osaan, jossa aina edellisen vaiheen aikana kerätyt tiedot lisätään aiemmin luotuun matriisiin.



Kuva 4: Cheng et al. (2012) lähestymistapa modularisointiin

Cheng et al. (2012) prosessi aloitetaan suunnitteluparametrien relevanttius DSM määrytyksestä. Se eroaa Williamsson & Sellgrenin (2016) mallista ottamalla tuoterakenteen lisäksi huomioon sekä funktionaalisuuden ja tuotantoprosessien Design matriisin. Williamsson & Sellgren (2016) pyrkivät löytämään rajoittavat tekijät suunnittelulle Modularity Indication Matriisiin (MIM). Kaikille DSM ratkaisuille tyypillistä on kuvata komponenttien vuorovaikutussuhteet matriiseihin +-merkeillä merkitsevään positiivista suhdetta ja -merkkiä kuvaamaan toistensa toimintaa haittaavaa suh-

detta. Useat ratkaisut hyödyntävät Pimmlerin ja Eppingerin (1994) kehittämää matriisiin luokittelujärjestelmää, jossa suhde annetaan $\{-2,2\}$ väliltä. Tällä mallilla -2 tarkoittaa, että vuorovaikutus on estettävä, 0 tarkoittaa ettei elementeillä ole vaikutusta toisiinsa ja +2 tarkoittaa vuorovaikutuksen olevan välttämätön toiminnan kannalta. Helmer et al. (2010) laajensivat tätä luokittelua ottamaan huomioon elementtien tilallisen jakautumisen tuotteen sisällä omassa lähestymistavassaan.

Toisessa vaiheessa Chengin et al. (2012), Williamssonin ja Sellgrenin (2016) ja Helmerin et al. (2010) esittämät lähestymistavat eroavat toisistaan suuremmin. (Cheng et al., 2012) määrittävät suunnitteluparametrien tuotantoprosessien samankaltaisuus DSM:n yhdistämällä aiemmin määriteltyyn suunnitteluparametrien DSM:n prosessireitit ja samankaltaisuussuhteen. Prosessireitti määritellään jokaiselle suunnitteluparametrille antamalla jokaiselle elementille arvo 0 tai 1 riippuen meneekö se prosessilinkin kautta. Seuraavaksi yhdistetään DSM:n suunnitteluparametrien keskinäiset suhteet. Tässä kohtaa on määritetty sekä osien kokonaisrelevanttiusuhde sekä samankaltaisuussuhde. Matriisit yhdistetään kattavaksi suunnitteluparametrien riippuvuus DMS:ksi, joka kuvaa kokonaisvaltaisesti suunnitteluparametrien välisiä yhteyksiä. (Cheng et al., 2012)

Williamssonin & Sellgrenin (2016) keskittyvät tässä vaiheessa erottamaan negatiivisia vaikutuksia muihin ratkaisuihin omaavat tekniset ratkaisut. Epähaluttujen ratkaisujen poistamiseksi MIM:n päälle lisätään toinen identtinen kolmiomatriisi. Lopuksi strategisesti mukautettu DSM lasketaan lisäämällä toisiinsa alkuperäinen ja strategiansiirto DSM. Kaikki negatiiviset suhteet ja niihin liittyvät ratkaisut poistetaan ja tyhjät solut pysyvät muuttumattomina. (Williamssonin & Sellgren, 2016)

Sekä Cheng et al. (2012) ja Williamsson ja Sellgren (2016) hyödyntävät symmetrisiä matriiseja, kun (Helmer et al. 2010) käyttävät epäsymmetristä matriisia. Jotta heidän lähestymistapansa toimisi, on tällä keinolla selvitetty matriisi data sovitettava, jotta matriisista saadaan taas symmetrinen. Helmer et al. (2010) hyödyntävät myös heidän erilaisen luokittelunsa mahdollistamaa perspektiivin kavennusta, jossa solussa esitetty vuorovaikutuksen tyyppit voidaan tiivistää neljästä yhteen.

Viimeisessä vaiheessa kaikki menetelmät ryhmittelevät tuotteen klustereihin eli mahdollisiin moduuleihin. Cheng et al. (2012) sekä Helmer et al. (2010) hyödyntävät tässä vaiheessa geneettisiä algoritmeja ja lyhimmän kuvauksen mallia (MDL). (Williamsson & Sellgren, 2016) eivät määritä mitä ryhmittelytyökalua heidän lähestymistapansa mukaisesti tulisi käyttää. MDL rakennetaan valitsemalla perusosaa kuvaava

termi ja sen lisäksi lisätermi, joka liitetään kuvaamaan moduulien sisäisiä yhteenso-
pimattomuuksia. Tulokseksi saadaan matriisi, joka on muotoa $m \times n_n$. Tässä matrii-
sissa m on ennalta päätetty klustereiden/ryhmien määrä ja n_n on elementtien määrä
DSM matriisissa. Chengin et al. (2012) mukaan Ericsson ja Erixon (1999) optimaali-
nen moduulien määrä minimaalisen kokoamisajan saavuttamiseksi on

$$m \leq \sqrt{n_n}. \quad (2)$$

(Helmer et al. 2010) jatkavat vielä prosessoimalla tulokset, joita algoritmi on saanut
ja korjaavat sekä parantavat niitä teknisen toteutettavuuden osalta. Heidän puoliau-
tomaattinen jälkiprosessointinsa keskittyy erityisesti vahvojen positiivisten suhteiden
tarkasteluun.

3.1 Indeksipohjaiset työkalut

Indeksipohjaiset työkalut ratkaisevat kysymyksen, missä kohtaa modulaarisuudesta
ja standardoinnista on hyötyä tuoteperheessä. Samankaltaisuusindeksit kuvaavat
tuoteperheen ”hyvyyttä” yhdestä tai useammasta näkökulmasta. Tyypillistä indek-
sipohjaisille työkaluille on sisällyttää Martinin ja Ishiin (2002) kehittämät General Va-
riety Indexi ja Coupling Indexi komponenttien luokitteluun. General Variety Indexi
(GVI) luodaan QFD matriisin pohjalta ja kuvaa, kuinka paljon variaatiota tietyllä kom-
ponentilla on tuoteperheen eri tuotteissa (Martin & Ishii, 2002). Coupling Indexi (CI)
kuvaa komponentin todennäköisyyttä tulla korvatuksi, joko ulkoisista tai sisäisistä
systä. Yhdistettynä nämä kaksi indeksiä kertovat yleisimmät komponentit systeemissä,
kuinka paljon komponenttien uudelleensuunnittelu vaikuttaa muihin kom-
ponentteihin ja mitkä komponentit todennäköisimmin vaihtuvat uuteen tuoteperheen
elinkaareen aikana. (Martin & Ishii, 2002)

(Simpson et al., 2012) mukaan suurin osa indeksipohjaisista työkaluista tarvitse-
vat/turvautuvat diskreetteihin komponenttien ja osien määrään (laskevat komponent-
tien määrän, joilla on sama muoto/koko, materiaali, kokoamistyyli). Joitakin indeksejä
on kehitetty arvioimaan parametristä variaatiota (variaatiota suunnitteluparametrien
asetuksissa tuoteperheen sisällä). Yksi tällainen menetelmä on tuoteperheen ran-
gaistusfunktio (PFPF), jota voidaan käyttää tuoteperheen optimoinnissa (Simpson et
al., 2012). PFPF:ä käytetään mittaamaan samankaltaisuuden puutetta/ tuoteperheen

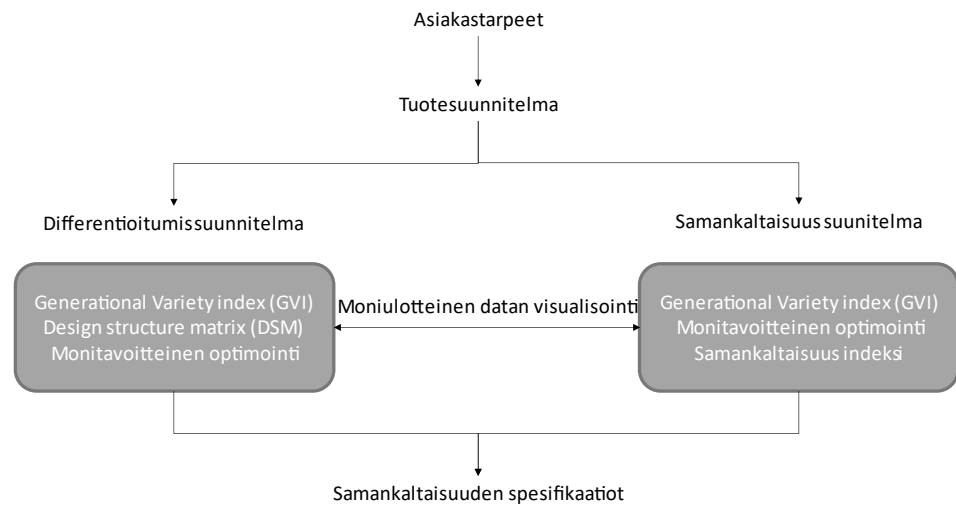
design muuttujien parametristen asetusten keskuudessa (Simpson et al., 2012). Toisin sanoen PFPF hylkii suunnitteluparametreja, jotka eivät ole yleisiä koko tuoteperheessä, samalla optimoiden haluttuja tavoitteita (Messac et al., 2002).

Indeksipohjaiset työkalut eivät vastaa yksin moduulien suunnittelun haasteisiin riittävän kokonaisvaltaisesti, jotta niitä voisi käyttää modularisointiin ilman muiden työkalujen tukea. Ne vaativat lähes aina rinnalleen jonkun muun työkalun, yleensä useita. Design Structure Matriisin ja optimointialgoritmi ovat tyypillisiä esimerkkejä tällaisista täydentävistä työkaluista. Puutteidensa takia indeksipohjaisten työkaluista julkaistujen tutkimusten määrä on huomattavasti pienempi kuin muilla työkaluilla. Tässä luvussa verrataan (Simpson et al., 2012), (Martin & Ishii, 2002) menetelmiä tuoteperheen kehityksessä, jossa hyödynnetään indeksipohjaisia menetelmiä.

Tuoteperheen kehitys

Pohjana työlleen (Simpson et al. 2012) hyödyntävät Robertsonin ja Ulrichin (1998) kehittämää ja yleistämää tuotealustan suunnittelun runkoa, joka koostuu kolmesta vaiheesta. Ensimmäisessä vaiheessa, jota kutsutaan tuotesuunnitelmaksi, määritetään, mitä tuotteita pitäisi tarjota milloinkin. Toisessa vaiheessa eli differentioitumissuunnitelmassa määritetään, mitkä tuotteet asetetaan millekin markkinasegmentille. Viimeinen eli kolmas vaihe on nimeltään samankaltaisuus suunnitelma ja sen tarkoituksena on määrittää mitkä osiot tuotteista pysyvät samoina tuoteperheen sisällä. (Simpson et al., 2012)

Kuvassa 5 esitetään kolmijakoinen Robertsonin ja Ulrichin (1998) tuoteperheen kehitysprosessi. Siihen on liitetty Simpsonin et al. (2012) lähestymistavan mukaisesti työkaluja, joita kussakin vaiheessa tulisi käyttää.



Kuva 5: Integroitu lähestymistapa tuoteperheen kehitykselle (Simpson et al. 2012)

Simpson et al. (2012) kritisoivat tätä mallia sen konkreettisten työkalujen puutteesta ja kehittivät mallia antamalla sille työkalut, mitä käyttää jokaisessa vaiheessa. Heidän mallinsa yhdistää markkinoiden segmentointiverkon, DSM:t, GVI, samankaltaisuusindeksit ja optimointia antaakseen asiakastarpeiden perusteella suosituksia, mitkä asiat pitäisivät olla yhteisiä ja mitkä tuotekohtaisia. (Simpson et al., 2012)

Ensimmäinen vaihe (Simpson et al., 2012) mallissa etenee lähes identtisesti (Martin, Ishii, 2002) mallin kanssa, johon se myös vahvasti pohjautuu. Myös Martin ja Ishii (2002) rakentavat GVI arvot ja Coupling Indeksien komponenteille. GVI generoimista varten Martin ja Ishii (2002) ottavat huomioon markkinasegmentin ja halutun tuotteen elinkaaren rakentaakseen niistä laadun talon matriisin. Seuraavaksi huomioidaan odotetut muutokset asiakastarpeissa, arvioidaan suunnittelutavoitteet tuotteelle. Aikaisemmin selvitettyjen tietojen perusteella saadaan aikaan GVI jokaiselle komponenteille.

Coupling indeksin rakentamiseen tarvitaan fyysinen komponenttien layout tuotteessa. (Martin, Ishii, 2002) ei määritä keinoa sen määrittämiseksi, mutta (Simpson et al., 2012) hyödyntää Design structure matriisia, johon generinen tuoterakenne markkinasegmentille voidaan hankkia esimerkiksi kilpailijoiden tuotteita tutkimalla. Layoutin jälkeen selvitetään komponenttien välillä kulkeva energia, materia, operaatiot, joita ne tarvitsevat toimiakseen. Näistä piirretään graafinen esitys. Lopuksi arvioidaan komponenttien herkkyyttä muutokselle. (Simpson et al., 2012)

Toisessa ja kolmannessa vaiheessa Martin & Ishii (2002) keskittyvät arvioimaan, mitkä komponentit tai alisysteemit tulisi priorisoida GVI ja CI arvojen perusteella eivätkä anna työkaluja komponenttien modularisointiin. Komponenttien standardointiin Martin ja Ishii (2002) kehottavat valitsemaan komponentit, joilla on suuret kehityskustannukset ja suuri GVI, koska ne todennäköisimmin muuttuvat asiakkaiden muuttuvien tarpeiden mukana. Simpson et al. (2012) lähestymistavassa alisysteemit, joilla on korkea GVI arvo eli korkea vaihtelevuus tuotteesta toiseen, on hyödyllistä modularisoida, kun taas alisysteemit, joilla on pieni GVI arvo, kannattaa standardoida. Lopuksi varmistetaan GVI ja DSM antamien tulosten oikeellisuus monitavoitteista optimoinnin, Product Family Penalty funktion ja moniulotteista datan visualisoinnin avulla (Simpson et al., 2012). Moniulotteisia visuaalisia työkaluja hyödynnetään näyttämään tuloksia ennen kuin mikään on valmista ja mahdollistamaan suunnittelijoiden muokkaavan suunnitelmaa ”lennossa” (Simpson et al., 2012).

4. YHTEENVETO

Tässä työssä tutkittiin, mitä modulaariset tuoterakenteet ovat. Huomattiin, että modulaarisuus on tärkeää yrityksille, jonka takia modulaarisista tuoterakenteista on paljon tehty paljon tutkimuksia ja lukuisat tutkijat ovat pyrkineet kehittää omanlaisiaan mallia havaitsemaansa tarkoitukseen. Tässä työssä esitettiin neljä eri luokkaa johon modularisoinnin lähestymistavat voidaan jakaa, perustana niiden erilainen lähestymistapa ositella tuotteet moduuleihin. Työssä esiteltiin myös suunnittelutyön onnistumisen kannalta tärkeä asiakaslähtöinen tuotekehitys. Asiakaslähtöisen tuotekehityksen tavoitteena on keskittää suunnittelutyö asiakkaan kannalta hyödyllisiin asioihin ja tällä tavalla varmistaa, että kehitetty tuote vastaa asiakkaan tarpeita.

Työssä havaittiin tiettyjä trendejä erilaisia lähestymistapoja tarkasteltaessa. Huomattiin, että funktionaalisen modularisoinnin hyödyntäminen lähestymistavoissa on vähentynyt viime vuosina. Sen sijaan tutkimustyö keskittyy hyödyntämään matriiseja ja algoritmeja. Sekä asiakaslähtöinen tuotekehitys, optimointipohjaiset lähestymistavat ja matriisipohjaiset lähestymistavat matriiseja moduulien määrittämisessä. Nykyajan trendi modularisoinnissa näyttääkin olevan Design structure matriisien yhdistäminen geneettisiin algoritmeihin optimaalisen tuoteratkaisun löytämiseksi.

Tutkimuksen tuloksena havaittiin hyvän nykyaikaisen modulaarisen tuoterakenteen kehitysprosessin piirteitä. Onnistunut tuoterakenteen kehitys tulisi sisältää asiakkaan tarpeita ja hänen arvostamiaan ominaisuuksia. Tällä alueella asiakaskeskeinen tuotekehityksen malli varmistaa sen, että asiakas on varmasti tuotteeseen tyytyväinen. Lukuisat lähestymistavat käyttivät kilpailijoiden tuotteita kehitettävän tuotteen suorituskyvyn vertaamiseen, sekä tuoterakenteen kehitykseen. Hyvän tuoterakenteen kehitysprosessin tulisi hyödyntää kilpailijoilta saatavaa tietoa. Kolmantena hyvän nykyaikaisen modulaarisen tuoterakenteen kehitysprosessin osana on matriisien hyödyntäminen tuotteen osittelussa moduuleihin. Matriisit voivat kuvata suhteiden tyyppiä ja voimakkuutta toisin kuin perinteiset funktionaaliset mallit ja ne voidaan syöttää tietokoneeseen. Neljäs hyvän nykyaikaisen modulaarisen tuoterakenteen kehitysprosessin osa on algoritmien hyödyntäminen matriisien käsittelyssä. Monimutkaisissa tuotteissa matriisit kasvavat suuriksi, jolloin niiden manuaalinen käsittelystä tulee vaikeaa. Algoritmien on mahdollista käydä suuri määrä vaihtoehtoja läpi lyhyessä ajassa, mikä vapauttaa suunnitteluresursseja muihin kehitystyön osa-alueisiin.

Työn aineisto on rajallinen, ja monta tutkijoiden kehittämää työkalua jäi esittelemättä. Tämä rajoittaa työn yleispätevyyttä, koska trendejä voi jäädä huomaamatta, kun esitellään vain muutama työkalu jokaisesta luokasta. Tämän tyylin etuna oli, että pystyttiin paremmin ja tarkemmin tutkimaan lähestymistapoja, jotka työssä käytettiin läpi. Tulevaisuudessa tutkimusta voitaisiin jatkaa tarkastelemalla lisää tässä tutkimuksessa käytettyä modularisoinnin lähestymistapoja jokaisesta kategoriasta mahdollisten nyt huomaamatta jääneiden havaitsemiseksi ja työssä löydettyjen hyvän nykyaikaisen modulaarisen tuoterakenteen kehitysprosessin piirteiden vahvistamiseksi. Tutkimus nojasi vahvasti Pakkasen et al. (2016) luokitukseen lähestymistavoista. Jatkotutkimus voisi tutkia aihetta käyttäen erilaista luokitusperiaatetta modularisoinnin lähestymistavoille, jolloin saadut tulokset voisivat muodostaa erilaisen kuvan kuin nyt.

LÄHTEET

- Andreasen, M.M. (2011), "45 Years with design methodology", *Journal of Engineering Design*, Vol. 22 No. 5, pp. 293–332.
- Binay Kumar, Ritesh Kumar Singh, Surendra Kumar 2018, "Genetic Algorithm-based Multi-criteria Approach to Product Modularization", *International Journal of Technology*, vol. 9, no. 4, pp. 291-319.
- Blees, C., Jonas, H., & Krause, D. (2010). Development of modular product families. In *DSM 2010: Proceedings of the 12th International DSM Conference*, Cambridge, UK, 22.-23.07. 2010.
- Cheng, Q., Zhang, G., Gu, P. & Shao, X. 2012, "A product module identification approach based on axiomatic design and design structure matrix", *Concurrent engineering, research and applications*, vol. 20, no. 3, pp. 185-194.
- Chen-Ritzo, C., Ervolina, T., Harrison, T.P. & Gupta, B. 2011, "Component rationing for available-to-promise scheduling in configure-to-order systems", *European Journal of Operational Research*, vol. 211, no. 1, pp. 57-65.
- Eilmus, S., Gebhardt, N., Rettberg, R. & Krause, D. 2012, "Evaluating a methodical approach for developing modular product families in industrial projects", *DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia*.
- Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications*. MIT press.
- Erixon, G. (1998), *Modular Function Deployment - A Method for Product Modularisation*, The Royal Institute of Technology, Stockholm, Sweden.
- Fellini, R., Kokkolaras, M. and Papalambros, P.Y. (2006), "Commonality Decisions in Product Family Design", In: Simpson, T.W., Siddique, Z. and Jiao, J.R. (Eds.), *Product Platform and Product Family Design*, Springer, pp. 158–185.
- Franceschini, F., Galetto, M., Maisano, D. & Mastrogiacomo, L. 2015, "Prioritisation of engineering characteristics in QFD in the case of customer requirements orderings", *International Journal of Production Research*, vol. 53, no. 13, pp. 3975-3988.
- Fujimoto, T. (2007), *Competing to Be Really, Really Good – The behind-the-Scenes Drama of Capability Building Competition in the Automobile Industry*, International House of Japan, Tokyo.
- Förg, A., Stocker, J., Kreimeyer, M. and Lienkamp, M. (2014), "Enabling modularisation potentials by standardized vehicle layouts", In: Laakso, M. and Ekman, K. (Eds.), *Proceedings of NordDesign 2014*, The Design Society, Espoo, Finland, pp. 754–764.

- Gardner, D. (2009) Mass customization : how build to order, assemble to order, configure to order, make to order, and engineer to order manufacturers increase profits and better satisfy customers . Cupertino, CA: Happy About.
- Garver, M.S. 2012, "Improving the house of quality with maximum difference scaling", *The International Journal of Quality & Reliability Management*, vol. 29, no. 5, pp. 576-594.
- Ghaffari, A. & Ulsoy, A.G. 2017, "Component Swapping Modularity for Distributed Precision Contouring", *IEEE/ASME transactions on mechatronics*, vol. 22, no. 6, pp. 2625-2632.
- Ginting, R., Tarigan, U. & Panjaitan, N. 2019, *The Application of Quality Function Deployment and Ergonomics: A Case Study for A New Product Design of A Texon Cutting Tool*, IOP Publishing.
- Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (2000). A method for architecting product platforms. *Research in engineering design*, 12(2), 61-72.
- Greve, E. & Krause, D. 2019, "Long-term effects of modular product architectures: An empirical follow-up study", *Procedia CIRP*, vol. 84, pp. 731-736.
- Gupta, S. & Jain, S.K. (2013) A literature review of lean manufacturing, *International Journal of Management Science and Engineering Management*, 8:4, 241-249, DOI: 10.1080/17509653.2013.825074
- Gupta, S., & Okudan, G. (2007). Modular design: A review of research and industrial applications. *IIE Annual Conference.Proceedings*, 1563-1568.
- Habib, T. & Komoto, H. 2014, "Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting", *Mechatronics (Oxford)*, vol. 24, no. 7, pp. 788-804.
- Helmer, R., Yassine, A., & Meier, C. (2010). Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design*, 21(6), 647-675.
- Holmqvist, T. (2004), *Managing Product Variety through Product Architecture*, Chalmers University of Technology.
- Jianxin J., Simpson T., & Siddique Z. (2007) Product family design and platform-based product development: a state-of-the-art review
- Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent manufacturing*, 16(3), 371-390.
- Juuti, T., Pakkanen, J. & Lehtonen, T. 2019, "Empirical Study of Good, Bad and Ugly Modular Engineering Solutions in Machinery Manufacturing Industry", *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 2981-2990.
- Kamrani, A.K. & Nasr, E.A. 2010, "Modular Design" in *Engineering Design and Rapid Prototyping*, eds. A.K. Kamrani & E.A. Nasr, Springer US, Boston, MA, pp. 57-86.

- Krause, D. & Eilmus, S. 2011, "Methodical Support for the Development of Modular Product Families" in *The Future of Design Methodology*, ed. H. Birkhofer, Springer London, London, pp. 35-45.
- Lehtonen, T. 2007, *Designing modular product architecture in the new product development*, Tampere University of Technology.
- Lindh, B.-E. (1992), *Scania and Its Vehicles, 1891-1991*, Streiffert.
- Martin, M.V. & Ishii, K. 2002, "Design for variety: developing standardized and modularized product platform architectures", *Research in engineering design*, vol. 13, no. 4, pp. 213-235.
- Messac, A., Martinez, M.P. & Simpson, T.W. 2002, "Introduction of a Product Family Penalty Function Using Physical Programming", *Journal of Mechanical Design*, vol. 124, no. 2, pp. 164-172.
- Otto, Kevin N., Wood, Kristin L. 2001, "Product Design", ISBN 0-13-021271-7, Prentice Hall, New Jersey.
- Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. (2007). *Engineering Design A Systematic Approach* (3rd ed. 2007.). Springer London. <https://doi.org/10.1007/978-1-84628-319-2>
- Pahl, G. and Beitz, W. (2013), *Engineering Design: A Systematic Approach*, Vol. 11, Springer Science & Business Media.
- Pakkanen, J., Juuti, T. & Lehtonen, T. 2016, "Brownfield Process: A method for modular product family development aiming for product configuration", *Design Studies*, vol. 45, pp. 210-241.
- Pimmler, T. U., & Eppinger, S. D. (1994). *Integration analysis of product decompositions*.
- Pine, B. J. (1993). *Mass customization* (Vol. 17). Boston: Harvard business school press.
- Rai, R., & Allada, V. (2003). Modular product family design: agent-based Pareto-optimization and quality loss function-based post-optimal analysis. *International Journal of Production Research*, 41(17), 4075-4098.
- Ramdas, K. & Randall, T. 2008, "Does Component Sharing Help or Hurt Reliability? An Empirical Study in the Automotive Industry", *Management science*, vol. 54, no. 5, pp. 922-938.
- Robertson, D., & Ulrich, K. (1998). Planning for product platforms. *Sloan management review*, 39(4), 19.
- Seiler, F.M., Greve, E. & Krause, D. 2019, "Development of a Configure-to-Order-Based Process for the Implementation of Modular Product Architectures: A Case Study", *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 2971-2980.

- Simpson, T.W., Bobuk, A., Slingerland, L.A., Brennan, S., Logan, D. & Reichard, K. 2012, "From user requirements to commonality specifications: an integrated approach to product family design", *Research in Engineering Design*, vol. 23, no. 2, pp. 141-153.
- Simpson, T., Siddique, Z., & Jiao, R. (2006). *Product Platform and Product Family Design Methods and Applications* (1st ed. 2006.). Springer US, pp. 1-17
- Spooner, J.G. 1998, "Fujitsu to Build to Demand; Company plans configure-to-order program to reduce notebook costs", *PC week (U.S.ed.)*, vol. 15, no. 12, pp. 43.
- Steward, D. V. (1981) The design structure system: A method for managing the design of complex systems. *IEEE transactions on engineering management*. [Online] EM-28 (3), 71–74.
- Stone, R. B., Wood, K. L., & Crawford, R. H. (2000). A heuristic method for identifying modules for product architectures. *Design studies*, 21(1), 5-31.
- Ulrich, K. T. & Eppinger, S. D. (2016) *Product design and development*. Sixth edition. New York, NY: McGraw-Hill Education.
- Williamsson, D. & Sellgren, U. (2016) An Approach to Integrated Modularization. *Procedia CIRP*. [Online] 50613–617.
- Winterkorn, M. and Pötsch, H.D. (2012), "Volkswagen Golf VII: Launch of a new era", Sardinia.