

Aki-Petteri Anttonen

**TUOTEPOHJAISEN OHJELMISTOKEHITYKSEN HAL-  
LINTA JA KEHITTÄMINEN ORGANISAATIOSSA**



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2023

# TIIVISTELMÄ

Anttonen, Aki-Petteri

Tuotepohjaisen ohjelmistokehityksen hallinta ja kehittäminen organisaatiossa

Jyväskylä: Jyväskylän yliopisto, 2023, 59 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Seppänen, Ville

Tässä pro gradu-tutkielmassa tutkittiin tuotepohjaisen ohjelmistokehityksen hallintaa ja kehittämistä organisaatiossa. Tutkimuksen kohteena oleva organisaation oli sisälogistiikan automaatiojärjestelmiä valmistava yritys Cimcorp Oy. Tutkimuksessa pyrittiin luomaan organisaation ohjelmistotuotelinjalle toimintamalli ja vastaamaan tutkimuskysymykseen: ”Millaisella toimintamallilla tuotepohjaista ohjelmistokehitystä voidaan toteuttaa organisaatiossa?”. Tutkittavan organisaation ohjelmistotuotelinja on suhteellisen uusi, joten sillä ei ole vielä selkeitä toimintatapoja tai prosesseja. Selkeällä toimintamallilla organisaation ohjelmistotuotelinjan toimintaa voitaisiin tehostaa ja selkeyttää. Ohjelmistotuotelinja on kustannustehokas keino valmistaa yksilöityjä ja laadukkaita ohjelmistotuotteita asiakkaille. Ohjelmistotuotelinjan avulla voidaan valmistaa laadukkaampia tuotteita nopeammin ja edullisemmin verrattuna yksittäisten tuotteiden kehittämiseen. Selkeällä toimintamallilla on suora vaikutus ohjelmistotuotelinjan tuotteiden laatuun. Tutkimus toteutettiin suunnittelutieteellisenä tutkimuksena. Tutkimuksessa luotiin toimintamalli organisaation ympäristön ja tietopohjan avulla. Tietopohjaa varten suoritettiin kirjallisuuskatsaus, jonka avulla perehdyttiin ohjelmistotuotelinjan teoriaan. Organisaation ympäristöä tutkittiin puolistrukturoiduilla haastatteluilla, joilla selvitettiin organisaation toimintatapoja ja käytäntöjä ohjelmistotuotelinjaan liittyen. Tutkimuksen lopputuloksena syntyneitä toimintamallia arvioitiin puolistrukturoiduilla haastatteluilla. Arvioinnin perusteella voidaan todeta, että luotu toimintamalli soveltuisi organisaation ohjelmistotuotelinjalle ja sillä voisi olla positiivisia vaikutuksia sen toimintaan. Luodun toimintamallin avulla voitaisiin siis toteuttaa tuotepohjaista ohjelmistokehitystä organisaatiossa.

Asiasanat: Ohjelmistotuotelinja, ohjelmistotuote, tuotepohjainen ohjelmistokehitys

## ABSTRACT

Anttonen, Aki-Petteri

Managing and developing product-based software development in an organisation

Jyväskylä: University of Jyväskylä, 2023, 59 pp.

Information Systems Science, Master's Thesis

Supervisor: Seppänen, Ville

This master's thesis examined the management and development of product-based software development in an organization. The organization under study was Cimcorp Oy, a company manufacturing automation systems for intralogistics. The research aimed to create a operating model for the software product line of the organization and to answer the research question "What kind of operating model can be used to implement product-based software development in an organization?". The software product line of the organization under study is relatively new, so it does not yet have a clear set of practices or processes. A clear approach could improve the efficiency and clarity of the organization's software product line. The software product line is a cost-effective means of producing customized, high-quality software products for customers. A software product line can produce higher quality products faster and at a lower cost than developing individual products. A clear operating model has a direct impact on the quality of the software product line's products. The study was carried out as a design science research. The study created an operating model using the organizational environment and knowledge base. For the knowledge base, a literature review was conducted to familiarize the theory of the software product line. The organizational environment was explored through semi-structured interviews to find out about the organization's policies and practices in relation to the software product line. Semi-structured interviews were used to evaluate the resulting operating model. The evaluation concluded that the policy would be suitable for the organization's software product line and could have a positive impact on its operations. It could therefore be used to implement product-based software development in the organization.

Keywords: Software product line, software product, product-based software engineering

## KUVIOT

KUVIO 1 Tietojärjestelmätieteen tutkimuksen viitekehys (Hevner, March, Park & Ram, 2008, s. 80).....	11
KUVIO 2 Tutkimuksen viitekehys (sovellettu Hevner ym., 2004.).....	13
KUVIO 3 Ohjelmistotuotelinjan keskeiset prosessit (Clements & Northrop, 2002, s. 30).....	16
KUVIO 4 Ohjelmistotuotelinjan ja yksittäisten tuotteiden kumulatiiviset kustannukset (Pohl, Böckle & Van Der Linden, 2005, s. 10).....	18
KUVIO 5 Ohjelmistotuotelinjan tuotteiden tuotantoaika verrattuna yksittäisiin tuotteisiin. (Pohl, Böckle & Van Der Linden, 2005, s. 11) .....	19
KUVIO 6 Ohjelmistotuotelinjan tuotantoprosessien viitekehys. (Van der Linden, Schmid & Rommes, 2007, s. 48) .....	22
KUVIO 7 Toimintamalli ensimmäisen iteraation jälkeen. ....	42
KUVIO 8 Toimintamalli toisen iteraation jälkeen.....	44
KUVIO 9 Toimintamalli kolmannen iteraation jälkeen. ....	45
KUVIO 10 Organisaation ohjelmistotuotelinjalle luotu toimintamalli.....	55

## TAULUKOT

Taulukko 1 Ydinkomponenttien kehittämisen prosessit. (Van der Linden, Schmid & Rommes, 2007).....	24
Taulukko 2 Tuotteen kehittämisen prosessit. (Van der Linden, Schmid & Rommes, 2007) .....	28
Taulukko 3 Yhteenveto toimintamallin arvioinnista .....	53

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO.....	7
2	TUTKIMUSMENETELMÄ .....	10
2.1	Suunnittelutieteellinen tutkimus tietojärjestelmätieteessä .....	10
2.2	Viitekehys sovellettuna tähän tutkimukseen .....	12
2.3	Ohjelmistotuotelinja tutkittavassa organisaatiossa .....	14
3	OHJELMISTOTUOTELINJA .....	15
3.1	Ohjelmistotuotelinjan perusteet .....	15
3.2	Ohjelmistotuotelinjan hyödyt .....	17
4	OHJELMISTOTUOTELINJAN PROSESSIT .....	21
4.1	Keskeiset prosessit .....	21
4.2	Ydinkomponenttien kehittäminen .....	23
4.2.1	Tuotteen hallinta.....	23
4.2.2	Vaatimusten määrittely .....	24
4.2.3	Arkkitehtuurin määrittely.....	25
4.2.4	Ydinohjelmistokomponenttien luominen tai hankinta .....	26
4.2.5	Testaaminen .....	26
4.3	Tuotteen kehittäminen .....	27
4.3.1	Vaatimusten määrittely .....	29
4.3.2	Arkkitehtuurin määrittely.....	29
4.3.3	Tuotteen luominen tai hankinta.....	29
4.3.4	Testaaminen .....	30
4.4	Johtaminen ja hallinta .....	30
5	HAASTATTELUIEN ANALYYSI.....	32
5.1	Organisaation ohjelmistotuotelinjan tavoitteet .....	32
5.2	Tunnistettuja haasteita .....	33
5.3	Toimintatapoja .....	35
6	TOIMINTAMALLIN SUUNNITTELU JA LUOMINEN.....	40
6.1	Toimintamallin tavoitteet .....	40
6.2	Toimintamallin luominen.....	41
6.2.1	Ensimmäinen iteraatio kirjallisuuden pohjalta.....	41
6.2.2	Toinen iteraatio haastattelujen analysoinnista.....	42
6.2.3	Kolmas iteraatio kirjallisuuden täydennyksistä .....	45
6.3	Toimintamalli .....	46

7	TOIMINTAMALLIN ARVIOINTI.....	49
7.1	Yleiset arviot ja mielipiteet .....	49
7.2	Kehitysideat, puuttuvat asiat ja muut kommentit.....	50
8	JOHTOPÄÄTÖKSET .....	54
	LÄHTEET .....	57

# 1 JOHDANTO

Yrityksillä on kasvava tarve valmistaa asiakkailleen yksilöityjä ja laadukkaita ohjelmistotuotteita, jotka täyttävät asiakkaiden tarpeet. Samaan aikaan ohjelmistotuotteiden laatuvaatimukset ja kompleksisuus ovat kasvaneet merkittävästi nykymarkkinoilla. Vastatakseen kasvaviin laatuvaatimuksiin ja muuttuvaan toimintaympäristöön, ohjelmistotuotteiden on oltava modulaarisia, joustavia ja niiden on oltava helposti laajennettavissa uusilla ominaisuuksilla. (Grüner, Burger, Kantonen & Rückert, 2020.) Ohjelmistotuotelinja on yksi tunnettu lähestymistapa tuotepohjaiseen ohjelmistokehitykseen, joka mahdollistaa laadukkaiden ohjelmistotuotteiden valmistamisen kustannustehokkaasti (Bockle, Clements, McGregor, Muthig & Schmid, 2004).

Ohjelmistotuotelinja on joukko ohjelmistotuotteita, jotka jakavat samoja ominaisuuksia ja toimintoja. Ohjelmistotuotelinjan tuotteet vastaavat tietyn markkinasegmentin vaatimuksia tai ne täyttävät tietyn tarpeen. (Northrop ym. 2007.) Ohjelmistotuotelinjan tuotteet ovat siis keskenään samankaltaisia, mutta sisältävät usein muuttuvia asiakaskohtaisia ominaisuuksia. Ohjelmistotuotelinjan toiminnassa keskeisessä osassa ovat ydinkomponentit. Ydinkomponentit ovat komponentteja, joita voidaan hyödyntää useissa ohjelmistotuotelinjan tuotteissa. (Van der Linden, Schmid & Rommes, 2007.) Ohjelmistotuotelinjan toiminta perustuu siis komponenttien systemaattiseen uudelleenkäyttöön.

Komponenttien systemaattinen uudelleenkäyttö ohjelmistotuotelinjassa mahdollistaa ohjelmistotuotteille korkeamman laadun, nopeamman toimitusajan ja alhaisemmat kustannukset verrattuna yksittäisten tuotteiden valmistamiseen (Metzger & Pohl, 2014). Ohjelmistotuotelinja on siis erinomainen keino tuottaa laadukkaita ja yksilöityjä tuotteita tietyn markkinasegmentin asiakkaille. Ohjelmistotuotelinjan toiminnassa on kuitenkin suuressa roolissa toimintamalli, jolla ohjelmistotuotelinjaa kehitetään (Van der Linden, Schmid & Rommes, 2007).

Tässä tietojärjestelmätieteen pro gradu-tutkielmassa tuotepohjaista ohjelmistokehitystä tarkastellaan Cimcorp Oy:n organisaatiossa. Cimcorp Oy on sisälogistiikan automaatiojärjestelmiä valmistava yritys. Organisaatiolla on suhteellisen uusi ohjelmistotuotelinja, jolle ei ole vielä määritelty selkeitä toimintatapoja.

Tämän tutkimuksen tarkoituksena on tutkia organisaation käytäntöjä ja toimintatapoja ja muodostaa organisaatiolle toimintamalli, jonka pohjalta ohjelmistotuotelinjan toimintaa voidaan ohjata ja kehittää. Selkeällä toimintamallilla on selvä vaikutus ohjelmistotuotelinjan tuotteiden laatuun (Van der Linden, Schmid & Rommes, 2007).

Tutkimus toteutettiin suunnittelutieteellisenä tutkimuksena (Design Science Research). Tutkimuksen tavoitteena oli toteuttaa organisaatiolle toimintamalli ohjelmistotuotelinjan toimintaan ja vastata samalla tutkimuskysymyksen:

- Millaisella toimintamallilla tuotepohjaista ohjelmistokehitystä voidaan toteuttaa organisaatiossa?

Toimintamallin luomisessa hyödynnetään puolistrukturoituja haastatteluja, joilla pyritään keräämään tietoa organisaation nykyisistä käytännöistä ja toimintatavoista. Toimintamallin luomisessa hyödynnetään myös ohjelmistotuotelinjan teoriaa, joka muodostaa tutkimuksen teoriapohjan. Teoriapohjaa varten toteutettiin kirjallisuuskatsaus.

Kirjallisuuskatsausta varten aineistoa etsittiin pääasiallisesti Google Scholar ja IEEExplore-palveluista. Lähdeaineistoksi valittiin tieteellisiä julkaisuja ja artikkeleita. Lähdeaineistoa etsiessä hakusanoina käytettiin esimerkiksi ”software product line”, ”software product line engineering”, ”ohjelmistotuotelinja” ja ”software product”. Lähdeaineistoa valittaessa pyrittiin huomioimaan aineiston luotettavuus ja ajankohtaisuus.

Toimintamallin luomisen jälkeen, toimintamallille suoritettiin vielä arviointi. Arvioinnilla pyrittiin selvittämään, soveltuuko toimintamalli organisaation ohjelmistotuotelinjalle. Arvioinnissa kysyttiin myös eri sidosryhmiltä, että käyttäisivätkö he toimintamallia ja muita mielipiteitä toimintamallista. Toimintamallin arviointiin hyödynnettiin puolistrukturoituja haastatteluja.

Tutkielman rakenne koostuu tiivistelmästä, sisällysluettelosta, johdannosta, sisältöluvuista, johtopäätöksistä ja lähdeluettelosta. Tutkimuksessa on kuusi sisältölukua. Sisältöluvuista yksi käsittelee tutkimuksen tutkimusmetodeja, kaksi lukua esittelee tutkimuksen teoriapohjan ja kolme lukua käsittelee tutkimuksen empiiristä osiota.

Ensimmäisessä sisältöluvussa esitellään tutkimuksessa käytetyt tutkimusmenetelmät. Luvun tarkoituksena esitellä ja perustella lukijalle tutkimuksen toteutukseen valitut tutkimusmenetelmät. Luvussa kerrotaan myös, miten tutkimusmenetelmät on sovellettu tähän tutkimukseen. Luvun lopussa esitellään tutkimuksen kohteena oleva organisaatio ja sen ohjelmistotuotelinja.

Ensimmäisessä teorialuvussa käydään läpi ohjelmistotuotelinjan perusteita. Luvun alussa käydään läpi keskeisiä asioita ohjelmistotuotelinjan toiminnasta ja esitellään ohjelmistotuotelinjaan liittyviä käsitteitä. Luvun lopussa käydään läpi ohjelmistotuotelinjalla saavutettavia hyötyjä.

Toisessa teorialuvussa perehdytään ohjelmistotuotelinjan toimintaan tarkemmin. Luvussa käydään läpi ohjelmistotuotelinjan toimintaan liittyviä proses-



seja. Tarkastelua on rajattu käsittelemään tarkemmin ohjelmistotuotelinjan teknistä puolta, minkä takia ohjelmistotuotelinjan johtamista ja hallinnointia on käsitelty vain pintapuoleisesti. Luvussa esitellään tarkasti ohjelmistotuotelinjan ydinkomponenttien ja tuotteen kehittämisen prosessit ja niihin liittyvät alaprosessit.

Empiirisen osion ensimmäinen luku käsittelee tutkimuksessa toteutettujen haastattelujen analysointia. Luvussa käydään läpi haastatteluissa esiin nousseita asioita. Aluksi käydään läpi mitä tavoitteita organisaation ohjelmistotuotelinjalle on, tämän jälkeen esitellään organisaatiossa esiintyviä haasteita ja lopuksi käydään läpi organisaation nykyisiä käytäntöjä ja toimintatapoja ohjelmistotuotelinjalle.

Empiirisen osion toisessa luvussa käydään läpi organisaation ohjelmistotuotelinjan toimintamallin luominen. Aluksi määritellään, mitkä ovat toimintamallin tavoitteet. Tämän jälkeen esitellään toimintamallin luominen kolmessa iteraatiossa. Ensimmäisessä iteraatiossa toimintamallille luodaan pohja teoriapohjan avulla. Toisessa iteraatiossa toimintamallia täydennetään haastattelujen analyysin pohjalta. Kolmannessa iteraatiossa hyödynnetään vielä teoriapohjaa ja lisätään toimintamalliin sieltä mahdollisesti puuttuvia elementtejä.

Empiirisen osion viimeisessä luvussa toimintamallia pyritään arvioimaan. Arviointia varten toteutettiin neljä haastattelua. Luvussa tuodaan esille haastateltavien mielipiteitä ja kommentteja toimintamallista sekä mahdollisia puutteita ja lisäyksiä.

## 2 TUTKIMUSMENETELMÄ

Tämä tutkimus toteutetaan suunnittelutieteellisenä tutkimuksena (Design Science Research). Tämän luvun tarkoituksena on esitellä tutkimuksessa hyödynnettävät menetelmät, sekä perustella niiden valinta. Aluksi käydään läpi suunnittelutieteellistä tutkimusta tietojärjestelmätieteessä ja esitellään tutkimuksessa hyödynnettävä viitekehys. Tämän jälkeen käydään tarkemmin läpi, miten viitekehystä on sovellettu tähän tutkimukseen. Lopuksi esitellään tutkimuksen kohteena olevaa organisaatiota ja ohjelmistotuotelinjaa.

### 2.1 Suunnittelutieteellinen tutkimus tietojärjestelmätieteessä

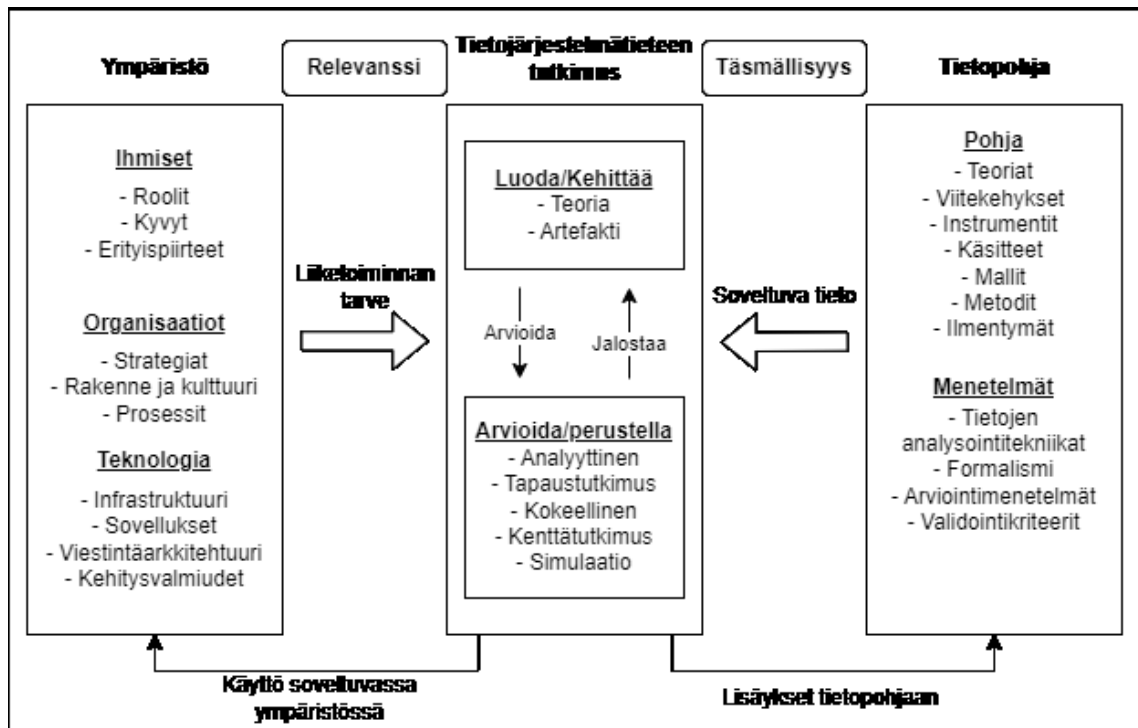
Suunnittelutieteellinen tutkimus tietojärjestelmätieteessä on tutkimusmenetelmä, jossa luodaan IT-artefakti ratkaisuna organisaation ongelmaan ja arvioidaan artefaktin toimintaa. IT-artefakteja voivat olla esimerkiksi toimintamallit, algoritmit tai prototyyppijärjestelmät. (Hevner, March, Park & Ram, 2004.) Suunnittelutieteellinen tutkimus pyrkii siis luomaan konkreettisen ratkaisun tunnistettuun ongelmaan ja arvioi ratkaisun toimivuutta.

Perinteisiin behaviorismiin suuntautuviin tutkimusmetodeihin verrattuna suunnittelutieteellinen suuntautuminen on melko uusi (Peffer, Tuunanen, Rothberger & Chatterjee, 2007). Behaviorismiin ja suunnittelutieteellisyyteen suuntautuvien tutkimusten keskeisenä erona on Hevnerin ym. (2004) mukaan se, että behaviorismi pyrkii selvittämään ilmiön tai teorian totuutta, kun taas suunnittelutieteellinen tutkimus pyrkii enemmän käytännöllisyyteen ja luomaan ratkaisua tunnistettuun ongelmaan.

Tämä tutkimus noudattaa Hevnerin ym. (2004) esittelemää viitekehystä tietojärjestelmätieteen suunnittelutieteelliselle tutkimukselle. Viitekehys esitetty kuviossa 1. Hevnerin ym. (2004) viitekehyksessä kuvataan suunnittelutieteellisen tutkimuksen ympäristöä. Tietojärjestelmätieteen tutkimuksessa ympäristö

tyypillisesti koostuu ihmisistä, organisaatiosta ja heidän käyttämistä teknologioista. Tutkimuksen ympäristö antaa tutkimukselle aiheen liiketoiminnan tarpeen tai ongelman mukaan. (Hevner ym., 2004.)

Ympäristön ongelmiin tai tarpeisiin vaikuttavat merkittävästi esimerkiksi ihmisten roolit ja kyvyt sekä organisaation strategiat, rakenne ja kulttuuri. Tarpeen tai ongelman tulisi myös olla yhteydessä olemassa oleviin ympäristön teknologioihin ja tukea näitä. (Hevner ym., 2004.) Tutkimuksen aiheen tulisi siis olla relevantti ympäristön tarpeisiin eli suunnittelutieteellisen tutkimuksen tulisi pyrkiä ratkaisemaan ympäristössä esiintyvä ongelma tai tarve.



KUVIO 1 Tietojärjestelmätieteen tutkimuksen viitekehys (Hevner, March, Park & Ram, 2008, s. 80)

Hevnerin ym. (2004) viitekehyksessä varsinainen tietojärjestelmätieteen tutkimus on jaettu kahteen vaiheeseen. Ensin luodaan tai kehitetään ratkaisu ja sen jälkeen pyritään arvioimaan ja perustelemaan sitä. Luotu ratkaisu voi olla esimerkiksi uusi artefakti tai teoria, joka pyrkii ratkaisemaan organisaation liiketoiminnallisen tarpeen tai ongelman. (Hevner ym., 2004.)

Tutkimuksen toteuttamisen apuna on tietopohja, josta hyödynnetään tutkimukseen soveltuvaa tietoa. Hevnerin ym. (2004) mukaan tämä tuo tutkimukseen täsmällisyyttä. Tietopohja tarjoaa tutkimukselle pohjan, jonka avulla tutkimus voidaan toteuttaa. Hevnerin ym. (2004) viitekehyksessä tietopohjaan sisältyvät pohja ja menetelmät. Pohja koostuu aikaisemmasta teoriasta, viitekehyksistä ja esimerkiksi erilaisista malleista. Näitä hyödynnetään tutkimuksen luomis- tai kehittämisympäristössä. Tietopohjan menetelmät tarjoavat puolestaan tutkimukselle

ohjeet ja työkalut arviointi- ja perusteluvaiheeseen. Menetelmiä voivat olla esimerkiksi erilaiset tietojen analysointitekniikat, arviointimenetelmät tai validointikriteerit. (Hevner ym., 2004.)

Kontribuutiona tietojärjestelmätieteen tutkimus voi tarjota lisäyksiä tietopohjaan. Jos toteutettu tutkimus tarjoaa esimerkiksi uuden teorian tai mallin, joka on todettu toimivaksi ja täsmälliseksi, voidaan se lisätä olemassa olevaan tietopohjaan. Tutkimus voi tarjota lisäksi organisaatiolle ratkaisun heidän liiketoiminnalliseen tarpeeseen tai ongelmaan. Tässä kuitenkin edellytyksenä, että tuotettu ratkaisu soveltuu ympäristöön, jossa sitä tulisi käyttää. (Hevner ym., 2004.) Toteutetun tutkimuksen tulee siis olla täsmällinen ja relevantti, jotta se on hyödyllinen organisaatiolle sekä tieteelle.

## 2.2 Viitekehys sovellettuna tähän tutkimukseen

Tämän tutkimuksen tavoitteena on muodostaa toimintamalli tuotepohjaisen ohjelmistokehityksen hallinnalle. Tutkimuksen aihe on saatu organisaatiolta, jolla on tarve kehittää omaa tuotepohjaisen ohjelmistokehityksen hallintaa uuden ohjelmistotuotelinjan osalta. Tutkimus toteutetaan suunnittelutieteellisenä tutkimuksena, koska suunnittelutieteellinen tutkimus soveltuu uuden toimintamallin luomiseen ja arviointiin. Tutkimuksen tutkimuskysymyksenä on ”millaisella toimintamallilla tuotepohjaista ohjelmistokehitystä voidaan toteuttaa organisaatiossa?”.

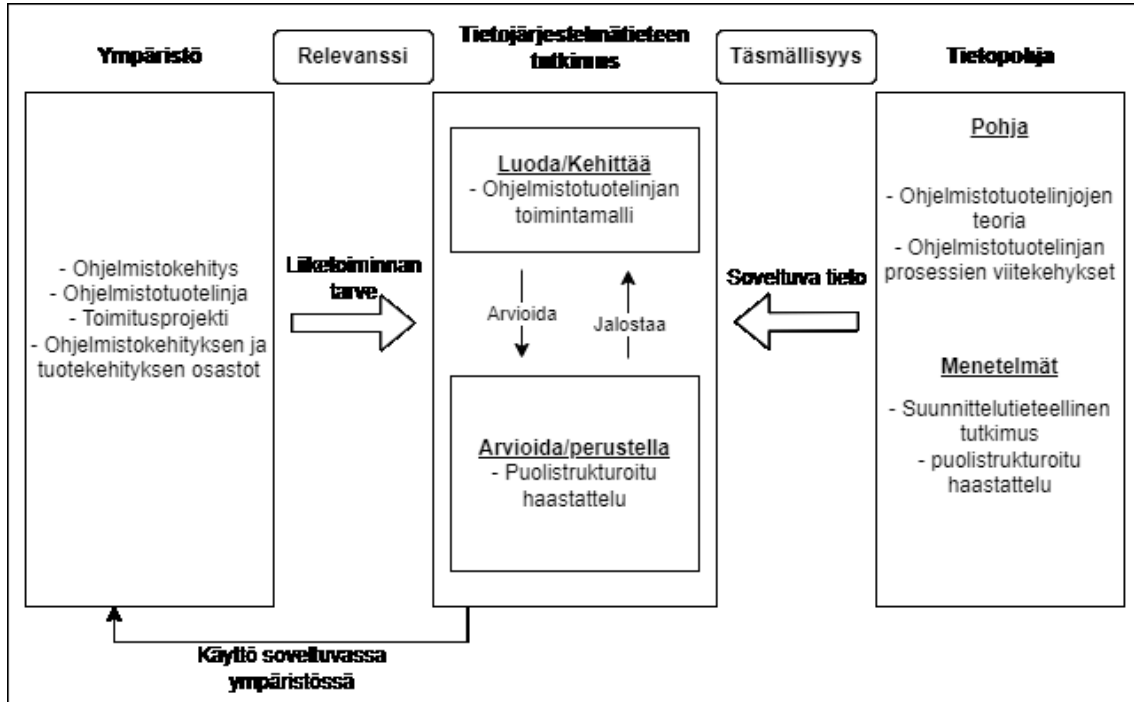
Tutkimuksen toteutuksessa hyödynnetään Hevnerin ym. (2004) esittelemää viitekehystä. Viitekehys sovellettuna tähän tutkimukseen esiteltynä kuviossa 2. Sovellettu viitekehys noudattaa pääpiirteittäin alkuperäistä viitekehystä. Tutkimuksen tavoitteena on luoda organisaatiolle toimintamalli, olemassa olevan teorian ja haastattelujen pohjalta.

Tutkimuksen viitekehyksessä ympäristö muodostuu organisaation ohjelmistokehitystyöstä, ohjelmistotuotelinjasta, toimitusprojekteista ja ohjelmistokehityksen ja tuotekehityksen osastoista. Organisaatiossa siis toteutetaan ohjelmistokehitystä toimitusprojekteja varten. Ohjelmistoa kehitetään ohjelmistotuotelinjan pohjalta. Organisaatiossa ollaan ottamassa käyttöön uutta ohjelmistotuotelinjaa ja tuotekehityksen sekä ohjelmistokehityksen osastoilla on liiketoiminnallinen tarve toimintamallille, joka ohjaa ja selkeyttää ohjelmistotuotelinjan toimintaa.

Varsinainen tutkimus toteutetaan alkuperäisen Hevnerin ym., (2004) esittelemän viitekehityksen mukaisesti kahdessa osassa. Tutkimuksessa pyritään luomaan toimintamalli sekä arvioidaan ja perustellaan sitä. Toimintamallin luomisessa hyödynnetään olemassa olevaa tietopohjaa ja hyödynnetään valittuja metodeja sen arviointiin ja jalostamiseen.

Tutkimuksen tietopohja muodostuu ohjelmistotuotelinjojen teoriasta ja ohjelmistotuotelinjan prosessien viitekehyksistä. Tietopohjan hankkimista varten toteutetaan kirjallisuuskatsaus. Kirjallisuuskatsauksen lähdeaineistoa etsitään

Google Scholar ja JYKDOK-palveluista. Hakusanoina käytetään esimerkiksi: "software product line", "software product line engineering", "software reuse" ja "ohjelmistotuotelinja". Lähdeaineistoa valittaessa pyritään huomioimaan niiden laatu ja luotettavuus. Lähdeaineisto koostuu lähinnä tieteellisistä artikkeleista ja julkaisuista.



KUVIO 2 Tutkimuksen viitekehys (sovellettu Hevner ym., 2004.)

Tietopohjan menetelmät osio koostuu suunnittelutieteellisestä tutkimuksesta, sekä puolistrukturoidusta haastattelusta. Tutkimus toteutetaan suunnittelutieteellisenä tutkimuksena, koska tarkoituksena on luoda artefakti, joka ratkaisee organisaation ongelman tai tarpeen. Puolistrukturoituja haastatteluja hyödynnetään toimintamallin arvioinnissa ja jalostamisessa.

Tutkimuksen yhdeksi metodiksi valittiin myös puolistrukturoitu haastattelu, koska se soveltuu hyvin avoimien kysymysten esittämiseen ja sen avulla saadaan yksittäisen henkilön mielipiteitä hyvin esille (Adams, 2015). Puolistrukturoitu haastattelu antaa haastateltavalle myös mahdollisuuden tuoda esille monenlaisia aiheeseen liittyviä teemoja ilman, että liian strukturoitu haastattelu rajaisi niitä (Harvey-Jordan & Long, 2001). Puolistrukturoidun haastattelun voidaan nähdä olevan lähellä normaalia keskustelua, mutta sillä on kuitenkin tietty tarkoitus ja tema (Kvale & Brinkmann, 2009). Haastattelun keskeiset teemat johdettiin kirjallisuuskatsauksen tuloksista. Haastattelukysymykset keskittyivät erityisesti ohjelmistotuotelinjan ydinkomponenttien ja tuotteen kehittämiseen sekä niihin liittyviin prosesseihin.

Haastateltaviksi tutkimusta varten valittiin sellaisia henkilöitä, jotka ovat keskeisissä rooleissa ohjelmistotuotelinjan toiminnassa. Haastateltavia valittiin

tuotekehityksen ja ohjelmistokehityksen osastoilta. Näin saatiin kerättyä mielipiteitä ohjelmistotuotelinjan eri sidosryhmiltä organisaation sisällä. Tuotekehityksen vastuulla on ohjelmistotuotelinjan pohjan eli ydinkomponenttien hallinta ja ohjelmistokehityksen vastuulla on tuotteiden valmistaminen ohjelmistotuotelinjasta. Haastatteluja hyödynnetään sekä toimintamallin luomiseen, että sen arviointiin.

### **2.3 Ohjelmistotuotelinja tutkittavassa organisaatiossa**

Organisaatio, jossa tutkimus suoritetaan, on Cimcorp Oy. Cimcorp on sisälogistiikan automaatiojärjestelmiä valmistava yritys. Cimcorp valmistaa automaatiojärjestelmiä rengas-, jakelu-, ja elintarviketeollisuuden yrityksille. Yritys toimittaa asiakkaille yleensä kokonaisen automaatiojärjestelmän, joka koostuu laitteista ja niitä ohjaavasta ohjelmistosta. Cimcorpin tapauksessa siis yksittäinen tuote on asiakkaalle toimitettava automaatiojärjestelmä.

Tässä tutkimuksessa tutkittava ohjelmistotuotelinja on WCS-ohjelmisto eli varastonhallintaohjelmisto. Tällä ohjelmistolla voidaan hallita yrityksen toimittamaa automaatiojärjestelmää. Organisaatio on toteuttanut lukuisia asiakasprojekteja, mutta tutkittava ohjelmistotuotelinja on kuitenkin vielä kehitysvaiheessa. Tutkittavaa ohjelmistotuotelinjaa on testattu muutamissa asiakasprojekteissa, mutta sen kehittämistyö jatkuu edelleen.

Cimcorpin organisaatiossa ohjelmistotuotelinjan kannalta keskeisimmät sidosryhmät ovat tuotekehitys- ja ohjelmistokehitysosasto eli projektiosasto. Tuotekehitysosaston vastuulla on ohjelmistotuotelinjan alustan eli sen ydinkomponenttien kehittäminen. Projektiosasto vastaa puolestaan tuotteiden valmistamisesta asiakastilauksiin. Ohjelmistotuotelinjan toiminnan ollessa vasta alkutekijöissään, selkeää työnjakoa tai sidosryhmiä ei kuitenkaan ole vielä määritelty. Nykyiset sidosryhmät ja työnjako ovat määräytyneet lähinnä aikaisempien asiakasprojektien toteutuksen pohjalta.

### 3 OHJELMISTOTUOTELINJA

Tässä luvussa käydään läpi ohjelmistotuotelinjan yleistä teoriaa. Luvun tarkoitus on perehdyttää lukija ohjelmistotuotelinjan käsitteeseen ja esitellä ohjelmistotuotelinjan toimintaa yleisesti. Aluksi esitellään ohjelmistotuotelinjan perusteet ja tämän jälkeen ohjelmistotuotelinjan hyötyjä.

#### 3.1 Ohjelmistotuotelinjan perusteet

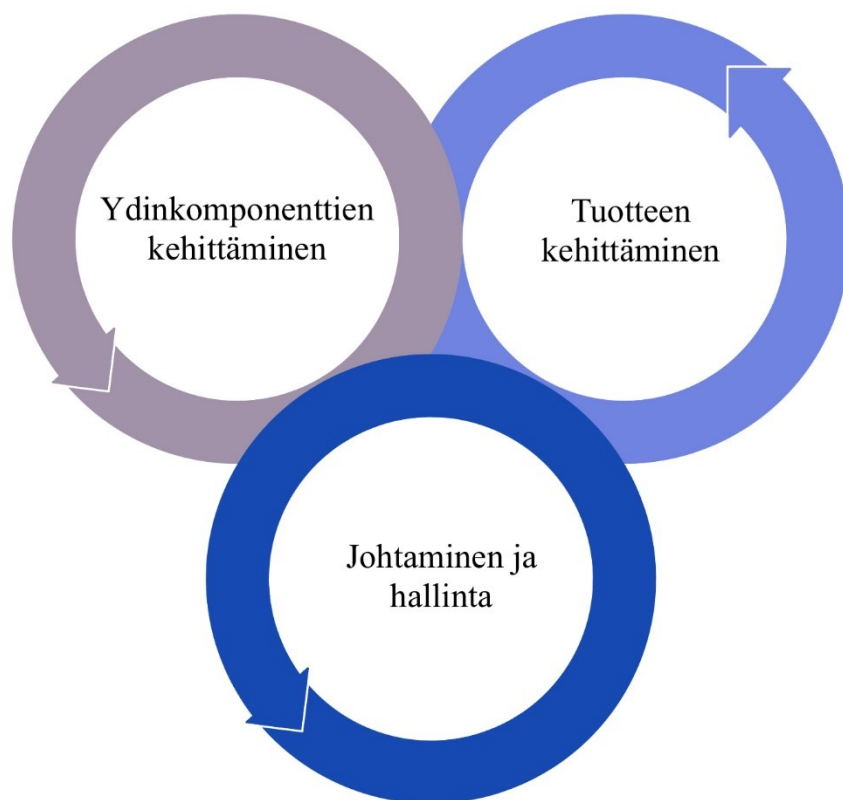
Ohjelmistotuotelinja on joukko ohjelmistotuotteita, jotka vastaavat tietyn markkinasegmentin tarpeita tai täyttävät tietyn tarpeen (Northrop ym. 2007). Ohjelmistotuotelinjan tuotteet ovat siis toiminnaltaan samankaltaisia tuotteita, jotka jakavat useita yhteisiä toiminnallisuuksia. Yksittäisissä tuotteissa voi kuitenkin olla eroavaisuuksia toiminnoissa, jotka vaihtelevat mahdollisen asiakkaan vaatimusten mukaan (Van der Linden, Schmid & Rommes, 2007).

Ohjelmistotuotelinjan tuotteet pohjautuvat aina tiettyihin ydinkomponentteihin. Ydinkomponentteja voivat olla esimerkiksi ohjelmistoarkkitehtuurit, uudelleenkäytettävät ohjelmistokomponentit, dokumentoinnit, testitapaukset tai prosessikuvaukset. Ohjelmistotuotelinjan tuotteet siis kootaan näistä yhteisistä ydinkomponenteista sekä tarvittavista tuotekohtaisista uusista komponenteista. Ydinkomponentit muodostavat kuitenkin koko ohjelmistotuotelinjan pohjan. (Northrop, 2002.) Ohjelmistotuotelinjan toiminta voidaan nähdä siis systemaattisena ohjelmistojen uudelleenkäyttönä (Bagheri, Ensan & Gasevic, 2012).

Ohjelmistotuotelinjasta saatavat hyödyt pohjautuvat ydinkomponenttien strategiseen ja suunniteltuun uudelleenkäyttöön. Kun organisaatio on muodostanut kattavan kokoelman ydinkomponentteja, voidaan jokaisen uuden tuotteen osalta saavuttaa merkittäviä säästöjä valmiina olevien ydinkomponenttien uudelleenkäytön myötä. (Clements & Northrop, 2002.) Organisaatiolla voi olla siis jo esimerkiksi entuudestaan valmiita ohjelmistokomponentteja, ohjelmistoarkkitehtuureja, vaatimuksia sekä prosesseja, joiden avulla uusia tuotteita voidaan

luoda. Tuotteiden yhtäläisyyksiä voidaan siis hyödyntää tuotannon kustannusten vähentämiseksi (Clements, 1999). Pääasiallisena tavoitteena ohjelmistotuotelinjalla on usein kehittää organisaation kykyä valmistaa tuotteita sekä parantaa tuotteiden laatua (Kang, Sugumaran & Park, 2009).

Ohjelmistotuotelinjan toiminnassa hyvin tärkeässä roolissa on prosessi, minkä avulla ohjelmistotuotelinjaa kehitetään. Hyvin määritetyllä tuotantoprosessilla on suuri vaikutus sen tuottamien tuotteiden laatuun (Van der Linden, Schmid & Rommes, 2007). Ohjelmistotuotelinjan tuotanto voidaan jakaa kahteen keskeiseen tuotantoprosessiin ydinkomponenttien kehittämiseen ja tuotteiden kehittämiseen (Van der Linden, Schmid & Rommes, 2007; Clements & Northrop, 2002). Ohjelmistotuotelinjan keskeiset prosessit kuvattuna kuviossa 3.



KUVIO 3 Ohjelmistotuotelinjan keskeiset prosessit (Clements & Northrop, 2002, s. 30)

Nämä prosessit eivät kuitenkaan ole täysin itsenäisiä, vaan ne ovat usein iteratiivisia prosesseja, jotka täydentävät toisiaan. Ydinkomponenttien kehittämisprosessi valmistaa ohjelmistotuotelinjalle alustan ja ydinkomponentit, jonka pohjalta tuotteen kehittämisprosessi valmistaa tuotteita. Tuotteen kehittämisprosessista annetaan palautetta ydinkomponenttien kehittämisprosessiin, jonka avulla ydinkomponentteja voidaan parantaa. (Van der Linden, Schmid & Rommes, 2007.) Ydinkomponenttien kehittäminen siis tarkastelee koko ohjelmistotuotelinjaa ja tuotteen kehittäminen keskittyy yksittäisiin tuotteisiin (Hallsteinsen, Hinchey, Park & Schmid, 2008).



Ohjelmistotuotelinjan toiminnassa keskeisenä osana on myös toiminnan johtaminen ja hallinta. Johtamisella on kriittinen osa ohjelmistotuotelinjan onnistumisessa. Osaavan johdon ja hallinnan avulla varmistetaan, että tuotantoprosesseilla on edellytykset toimia tehokkaasti ja prosessit toimivat oikeiden aktiviteettien parissa. (Clements & Northrop, 2002.) Johtamisen ja hallinnan vastuulla on myös varmistaa, että organisaation rakenne on ohjelmistotuotelinjalle soveltuva. Organisaation rakenteella on merkittäviä vaikutuksia ohjelmistotuotelinjan toimintaan. (Bosch, 2001.)

## 3.2 Ohjelmistotuotelinjan hyödyt

Ohjelmistotuotelinjan mahdollisina vaihtoehtoina organisaatiolla on yleisesti joko luoda tuotteet alusta asti yksittäisinä tuotteina tai luoda yksi yleinen tuote, joka pyrkii palvelemaan mahdollisimman monien asiakkaiden tarpeita. Näihin verrattuna ohjelmistotuotelinja voi tarjota organisaatiolle lukuisia erilaisia hyötyjä. (Apel, Batory, Kästner & Saake, 2016.)

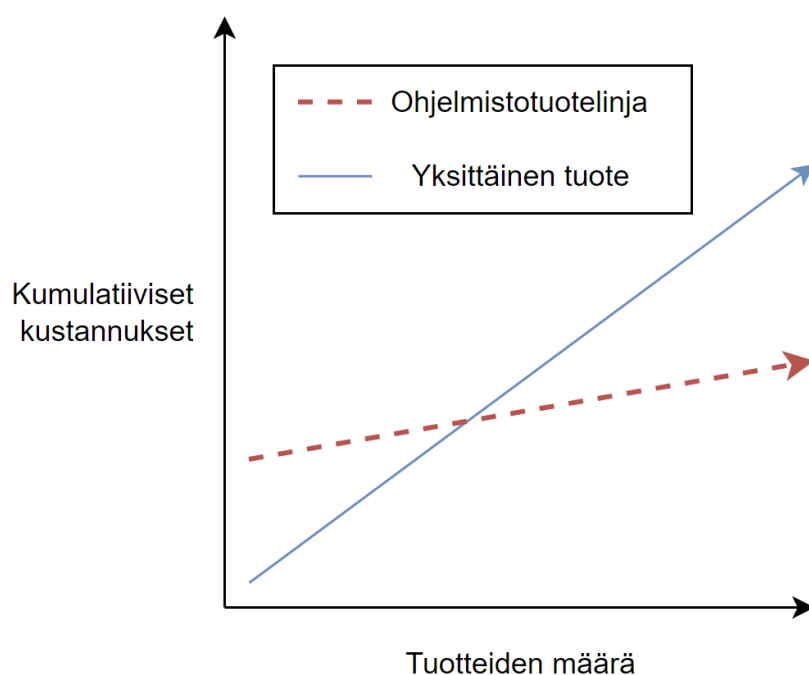
Ohjelmistotuotelinjan tarjoamia hyötyjä voivat olla esimerkiksi alhaisemmat kustannukset, korkeampi laatu ja nopeampi tuotantoaika (Metzger & Pohl, 2014). Useimmille organisaatioille nämä tehokkuuden ja kustannustehokkuuden parannukset ovatkin suurin motivaatio ohjelmistotuotelinjan aloittamiselle. Organisaatio voi myös tavoitella parempaa asiakastyytyväisyyttä ja tuotteen luotettavuutta laadukkaampien tuotteiden avulla. (Van der Linden, Schmid & Rommes, 2007.)

Ohjelmistotuotelinjan hyötyjen saavuttamiseksi organisaatiolta vaaditaan kuitenkin usein aluksi suuria investointeja ohjelmistotuotelinjaan (Kang, Sugumaran & Park, 2009). Ohjelmistotuotelinjan korkeat alkukustannukset johtuvat ydinkomponenttien luomisesta aiheutuvista kehityskustannuksista (Sugumaran, Park & Kang, 2006). Ohjelmistotuotelinjan toiminta pohjautuu uudelleen käytettäviin laadukkaisiin ydinkomponentteihin, joten organisaation tulee panostaa näiden kehitystyöhön heti ohjelmistotuotelinjan alkuvaiheessa.

Ohjelmistotuotelinjan hyödyt alkavat konkretisoitua, kun ydinkomponenttien pohjalta kyetään valmistamaan laadukkaampia tuotteita nopeammin ja edullisemmin (Pohl & Metzger, 2018). Ohjelmistotuotelinjan perustamisessa on kuitenkin aina riskinsä. Mahdollista on, että alkuinvestointien jälkeen todetaan, että ohjelmistotuotelinja ei sovellukaan organisaation tuotteiden tai toimialan tarpeisiin (Tüzün, Tekinerdogan, Kalender & Bilgen, 2015). Tässä tilanteessa kaikki organisaation investoinnit ohjelmistotuotelinjaan ovat olleet turhia. Ohjelmistotuotelinjan perustaminen tuleekin tästä syystä suunnitella erittäin tarkasti. (Schmid, 2001.)

Ohjelmistotuotelinjan aloittamisen onnistuessa alkuinvestoinnit maksavat kuitenkin itsensä takaisin, kun tuotteiden määrää lisätään verrattuna siihen, että tuotteet valmistettaisiin alusta alkaen yksittäisinä tuotteina. Pohlin, Böcklen ja Van Der Lindenin (2005) mukaan ohjelmistotuotelinjan kumulatiiviset kustan-

nukset ovat edullisemmat noin kolmen kehitetyn tuotteen jälkeen verrattuna siihen, että tuotteet valmistettaisiin yksittäisinä tuotteina. Tähän kuitenkin vaikuttavat merkittävästi organisaation rakenne, toimiala ja ohjelmistotuotelinjan alkuinvestointien määrä. (Pohl, Böckle & Van Der Linden, 2005.) Ohjelmistotuotelinjan ja yksittäisten tuotteiden kumulatiiviset kustannukset kuvattuna kuviossa 4.

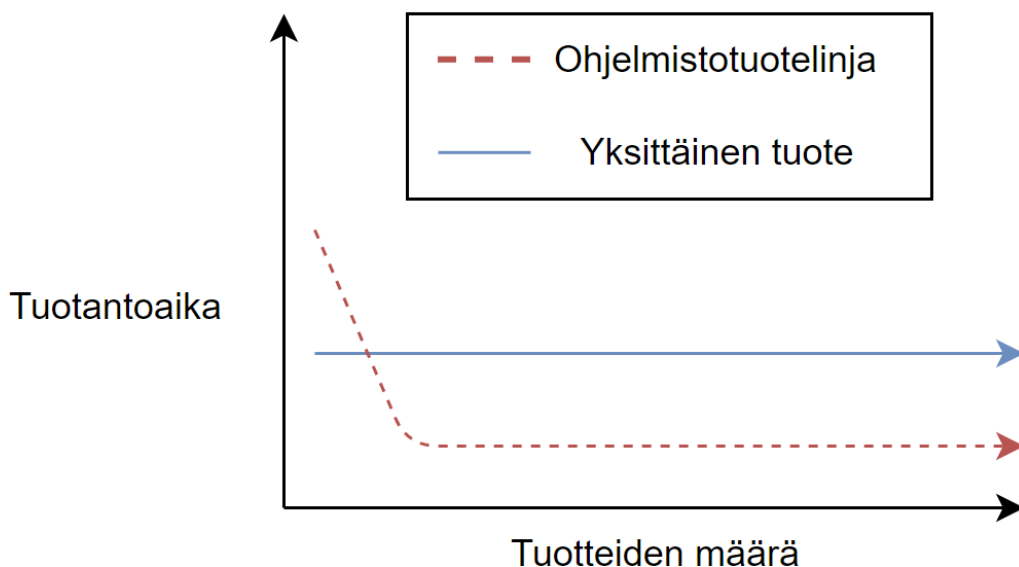


KUVIO 4 Ohjelmistotuotelinjan ja yksittäisten tuotteiden kumulatiiviset kustannukset (Pohl, Böckle & Van Der Linden, 2005, s. 10)

Kun ohjelmistotuotelinjalle on muodostettu laadukkaat ydinkomponentit, voi organisaatio alkaa valmistaa tuotteita näiden pohjalta. Tässä vaiheessa korkeat alkuinvestoinnit alkavat maksaa itseään takaisin, kun tuotteiden luominen ydinkomponenttien pohjalta ei vaadi niin paljoa aikaa ja resursseja, kuin niiden luominen tyhjästä vaatisi. (Pohl, Böckle & Van Der Linden, 2005.) Tämä johtuu siitä, että suuri osa tuotteesta voidaan toteuttaa ydinkomponenttien pohjalta vain pienillä muutoksilla. Tuotteisiin joudutaan kuitenkin usein tekemään uusia tuotekohtaisia ominaisuuksia, joten koko tuotetta ei välttämättä kyetä valmistamaan vain ydinkomponenttien pohjalta (Apel ym. 2016).

Ohjelmistotuotelinjan ydinkomponenttien uudelleenkäyttö mahdollistaa pienempien kehityskustannuksien lisäksi myös nopeamman tuotantoajan (Pohl, Böckle & Van Der Linden, 2005). Ohjelmistotuotelinjan toiminnassa kuitenkin ensimmäisten tuotteiden tuotantoaika on pidempi verrattuna yksittäisten tuotteiden luomiseen (Knauber, Bermejo, Böckle, Leite, van der Linden, Northrop & Weiss, 2002). Tämä johtuu suuresta alkuinvestoinnista ydinkomponenttien kehittämiseen. Kuitenkin samoin kuin kustannusten kanssa, tuotantoajat lyhenevät,

kun ohjelmistotuotelinjan pohjalta on luotu useampia tuotteita. (Pohl, Böckle & Van Der Linden, 2005.) Ohjelmistotuotelinjan tuotteiden tuotantoajat verrattuna yksittäisten tuotteiden tuotantoaikoihin kuvattuna kuviossa 5.



KUVIO 5 Ohjelmistotuotelinjan tuotteiden tuotantoaika verrattuna yksittäisiin tuotteisiin. (Pohl, Böckle & Van Der Linden, 2005, s. 11)

Nopeamman tuotantoajan ja matalampien kustannuksien lisäksi ohjelmistotuotelinja voi tarjota yritykselle laadukkaampia tuotteita (Montagud, Abrahão & Insfran, 2012). Kun tuotteita rakennetaan ohjelmistotuotelinjan ydinkomponenttien pohjalta, nämä ydinkomponentit ovat käyneet läpi testausprosessin jokaisen tuotteen valmistamisen yhteydessä (Pohl, Böckle & Van Der Linden, 2005). Näin pidetään huolta siitä, että ydinkomponentit ovat laadukkaita ja toimivat oikein. Lukuisien testauksien yhteydessä suurin osa virheistä ja ongelmista on huomattu. Tämän takia ohjelmistotuotelinjan tuotteet voivat olla parempilaatuisia kuin alusta asti yksittäisenä tuotteena valmistettu tuote. (Pohl, Böckle & Van Der Linden, 2005.)

Korkealaatuisempien tuotteiden lisäksi ohjelmistotuotelinja voi myös parantaa asiakkaiden tyytyväisyyttä tuotteisiin (Jones & Soule, 2002). Parempi asiakastytyväisyys pohjautuu pitkälti parempilaatuisen tuotteeseen. Laadukkaammasta tuotteesta löytyy vähemmän virheitä, jotka vaatisivat yhteydenottoja asiakaspalveluun tai aiheuttaisivat ongelmia tuotteen käyttämisessä ja tuottaisivat sitä kautta kustannuksia asiakkaalle. (Clements & Northrop, 2002.)

Ohjelmistotuotelinja tarjoaa hyötyjä myös ohjelmistotuotelinjan toiminnassa mukana oleville työntekijöille (Clements & Northrop, 2002.) Ohjelmistotuotelinjalla on selviä vaikutuksia esimerkiksi ohjelmistokehittäjän työhön. Ohjelmistotuotelinja poistaa ohjelmistokehittäjien työstä toistoa, kun tuotteiden keskinäiset yhtäläisyydet ovat toteutettuina ydinkomponenteissa. Samoja ominaisuuksia tai muuta samaa työtä ei siis tarvitse tehdä uudelleen jokaista tuotetta

varten. (Clements & Northrop, 2002.) Ohjelmistokehittäjä voikin keskittyä jokaisen tuotteen osalta vain niiden uniikkeihin asiakaskohtaisiin vaatimuksiin ja niiden toteuttamiseen. Tämä voi vaikuttaa merkittävästi työntekijän työmoraaliin ja tyytyväisyyteen (Clements & Northrop, 2002).

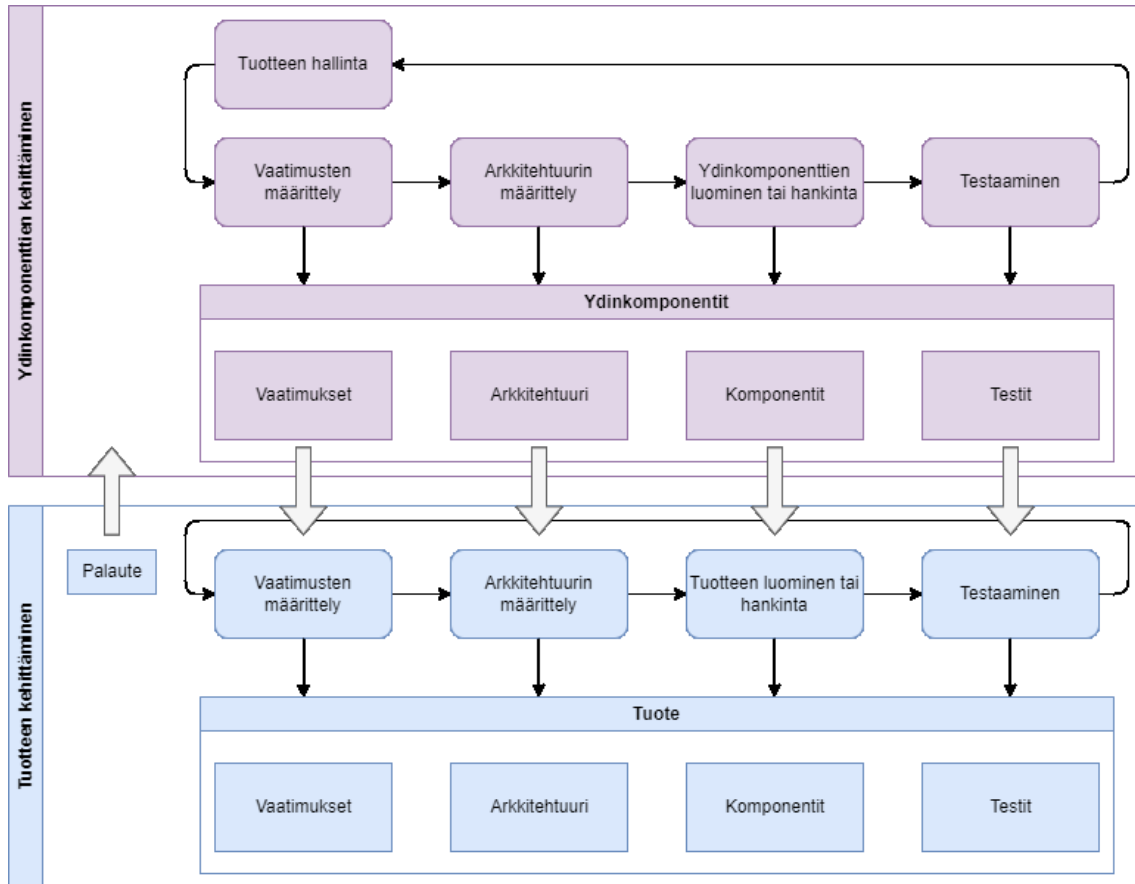
## 4 OHJELMISTOTUOTELINJAN PROSESSIT

Tässä luvussa esitellään ohjelmistotuotelinjan keskeiset prosessit. Luvun tarkoituksena on perehdyttää lukijalle ohjelmistotuotelinjaan liittyvät keskeiset prosessit. Luvussa käydään läpi keskeisiä toimintoja, joita prosesseihin liittyy. Aluksi käydään läpi prosessit yleisesti ja tämän jälkeen käydään jokainen prosessi tarkemmin läpi.

### 4.1 Keskeiset prosessit

Ohjelmistotuotelinjan toiminnassa keskiössä on kaksi tuotantoprosessia ja näiden tuotantoprosessien johtaminen ja hallinta. Ohjelmistotuotelinjan tuotantoprosessit ovat ydinkomponenttien kehittäminen ja tuotteen kehittäminen. (Clements & Northrop, 2002.) Ydinkomponenttien kehittämisestä käytetään joskus myös termiä kohdealueen kehittäminen (domain engineering) (Van der Linden, Schmid & Rommes, 2007). Tässä tutkimuksessa ohjelmistotuotelinjan prosesseja tarkastellaan pitkälti Clementsin ja Northropin (2002) ja Van der Lindenin, Schmidin ja Rommesin (2007) teoksien pohjalta. Nämä ovat ohjelmistotuotelinjan teorian keskeisiä ja laajalti tunnettuja teoksia, joita hyödynnetään uusissakin tutkimusartikkeleissa. Van der Lindenin, Schmidin ja Rommesin (2007) viitekehys ohjelmistotuotelinjan tuotantoprosesseista esitettyinä kuviossa 6.

Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä ydinkomponenttien kehittäminen on jaettu viiteen alaprosessiin. Nämä prosessit ovat tuotteen hallinta, ydinkomponenttien vaatimusten määrittely, ydinkomponenttien arkkitehtuurin määrittely, ydinkomponenttien luominen ja ydinkomponenttien testaaminen. Näiden alaprosessien lopputuloksena syntyy ohjelmistotuotelinjan ydinkomponentit, joita ovat esimerkiksi vaatimukset, arkkitehtuurit, ohjelmistokomponentit ja testit. Nämä ydinkomponentit muodostavat ohjelmistotuotelinjan alustan. (Van der Linden, Schmid & Rommes, 2007.)



KUVIO 6 Ohjelmistotuotelinjan tuotantoprosessien viitekehys. (Van der Linden, Schmid & Rommes, 2007, s. 48)

Tuotteen kehitysprosessi on Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä jaettu neljään alaprosessiin, jotka ovat tuotteen vaatimusten määrittely, tuotteen arkkitehtuurin mallintaminen, tuotteen luominen ja tuotteen testaaminen. Tuotteen kehitysprosessissa hyödynnetään ydinkomponentteja, joista luodaan ohjelmistotuotelinjan tuotteita. (Van der Linden, Schmid & Rommes, 2007.)

Monet organisaatiot lähestyvät ohjelmistotuotelinjaa proaktiivisella lähestymistavalla (Northrop, 2002). Proaktiivisessa lähestymistavassa ohjelmistotuotelinja aloitetaan kehittämällä ensin ydinkomponentit (Krueger, 2002). Organisaatiolla ei siis välttämättä ole tässä kohtaa vielä yhtäkään tuotetta, vaan kehitys aloitetaan puhtaasti määrittelemällä ohjelmistotuotelinjan kohde ja sen pohjalta aloitetaan ydinkomponenttien luominen. Toinen lähestymistapa ohjelmistotuotelinjaan on reaktiivinen (Northrop, 2002). Tässä lähestymistavassa organisaatiolla voi olla yksi tai useampi tuote valmiina, joiden pohjalta valmistetaan ohjelmistotuotelinjan ydinkomponentit.

## 4.2 Ydinkomponenttien kehittäminen

Ohjelmistotuotelinjan toiminnan kannalta keskeisenä tuotantoprosessina on ydinkomponenttien kehittäminen. Ydinkomponenttien kehittämisessä keskeisenä tehtävänä on luoda ja kehittää ohjelmistotuotelinjan ydinkomponentteja ja varmistaa, että näiden pohjalta kyetään valmistamaan ohjelmistotuotelinjan tuotteita. (Clements & Northrop, 2002.) Ydinkomponentit muodostavat koko ohjelmistotuotelinjan alustan.

Ydinkomponenttien kehittäminen myös määrittää ohjelmistotuotelinjan tuotteiden laajuuden ja kohdealueen sekä varmistaa, että ohjelmistotuotelinjan alustassa on riittävästi variaatiota, kattamaan kohdealueen määrittämät tuotteet (Van der Linden, Schmid & Rommes, 2007). Ydinkomponenttien kehittämisessä tulee huomioida myös olemassa olevien ydinkomponenttien ylläpito ja jatkokehitys (Reinhartz-Berger, Cohen, Bettin, Clark & Sturm, 2013).

Ohjelmistotuotelinjan ydinkomponentit muodostuvat erilaisista komponenteista, joita voidaan käyttää uudelleen ohjelmistotuotelinjan eri tuotteissa. Ydinkomponentteja voivat olla esimerkiksi vaatimukset, arkkitehtuurit, ohjelmistokomponentit ja testit. (Van der Linden, Schmid & Rommes, 2007.) Käytännössä siis mikä tahansa uudelleen käytettävä organisaatiolle hyödyllinen dokumentti, ohjelmisto, prosessi tai yksittäinen koodi voi olla ohjelmistotuotelinjan ydinkomponentti. Näitä komponentteja kehittäessä tulee kiinnittää erityisesti huomiota niiden uudelleenkäytettävyyteen (Schaefer & Hähnle, 2011).

Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä ydinkomponenttien kehittäminen koostuu viidestä eri alaprosessista. Jokaisella näistä on merkittävä rooli ohjelmistotuotelinjan ydinkomponenttien kehittämisessä. Viitekehyksen alaprosessit ovat tuotteen hallinta, ydinkomponenttien vaatimusten määrittely, ydinkomponenttien arkkitehtuurin määrittely, ydinkomponenttien luominen ja ydinkomponenttien testaaminen. (Van der Linden, Schmid & Rommes, 2007.) Ydinkomponenttien kehittämisen alaprosessit Van der Lindenin, Schmidin ja Rommesin (2007) mukaan tiivistettynä taulukossa 1.

### 4.2.1 Tuotteen hallinta

Tuotteen hallinnalla tarkoitetaan ohjelmistotuotelinjan skaalan ja kohdealueen hallintaa. Tuotteen hallinta siis määrittelee, minkälaisia tuotteita ohjelmistotuotelinjasta voidaan valmistaa, mitkä ovat tuotteen kohdemarkkinat ja mitä ominaisuuksia ja toimintoja tuotteessa on. (Van der Linden, Schmid & Rommes, 2007.) Tuotteen ominaisuuksiin vaikuttavat merkittävästi esimerkiksi organisaation toimiala, markkinat sekä toimialan standardit (Clements & Northrop, 2002). Tuotteen ominaisuuksien osalta on tärkeää määrittää, mitkä ominaisuudet ovat yhteisiä kaikkien ohjelmistotuotelinjan tuotteiden välillä ja mitkä ovat tuotekohdittaisia variaatioita. Tuotteen hallinnassa määritellään myös ohjelmistotuotelinjan markkinastrategia (Van der Linden, Schmid & Rommes, 2007).

Taulukko 1 Ydinkomponenttien kehittämisen prosessit. (Van der Linden, Schmid &amp; Rommes, 2007)

Prosessi	Keskeinen sisältö
Tuotteen hallinta	Ohjelmistotuotelinjan tuotteiden, kohdemarkkinoiden ja tarvittavien ominaisuuksien määrittely.
Vaatimusten määrittely	Selvitetään ohjelmistotuotelinjan sidosryhmien tarpeet ja määritetään niiden pohjalta tarkat vaatimuksen ohjelmistotuotelinjan ydinkomponenteille.
Arkkitehtuurin määrittely	Määritellään ohjelmistotuotelinjan ydinkomponenttien arkkitehtuuri eli määritellään se millaisia ohjelmistokomponentteja tarvitaan ja millaisia variaatiopisteitä niissä tulisi olla.
Ydinohjelmistokomponenttien luominen tai hankinta	Luodaan tai hankitaan tarvittavat ydinohjelmistokomponentit. Mahdollisuuksina luoda itse, ulkoistaa luominen, ostaa valmis komponentti tai etsiä olemassa olevista komponenteista sopiva.
Testaaminen	Testataan ydinkomponentit ja varmistetaan, että ne täyttävät niille annetut vaatimukset.

#### 4.2.2 Vaatimusten määrittely

Toinen alaprosessi ydinkomponenttien kehittämisessä on vaatimusten määrittely. Vaatimusten määrittelyssä määritetään, mitä vaatimuksia ohjelmistotuotelinjan ydinkomponenteille on. Ydinkomponenttien vaatimusten tulisi kattaa kaikkien ohjelmistolinjan mahdollisten tuotteiden vaatimukset. (Van der Linden, Schmid & Rommes, 2007.)

Vaatimusten määrittelyssä eri sidosryhmien tarpeilla on merkittävä rooli. Ydinkomponenttien vaatimuksia pohdittaessa sidosryhminä toimivat varsinaisten asiakkaiden eli lopputuotteen käyttäjien lisäksi myös esimerkiksi organisaation eri tiimit (Van der Linden, Schmid & Rommes, 2007). Esimerkiksi yksi tärkeä sidosryhmä on tuotteen kehittämiseen osallistuvat henkilöt. Vaatimuksissa tulisi huomioida tuotteen kehittämisen kannalta keskeisiä ominaisuuksia kuten esimerkiksi ydinkomponenttien käytön tehokkuus tai niiden laatu (Van der Linden, Schmid & Rommes, 2007).

Van der Linden, Schmid ja Rommes (2007) jakavat ydinkomponenttien vaatimusten määrittelyn viiteen eri vaiheeseen. Ensimmäinen vaihe on vaatimusten selvittäminen. Vaatimuksia voidaan selvittää analysoimalla eri sidosryhmien tarpeita. (Van der Linden, Schmid & Rommes, 2007) Vaatimuksia selvittäessä tulisi



erityisesti kiinnittää huomiota variaatiopisteisiin, joita voi esiintyä ydinkomponenteissa. Tarkastelussa tulisi myös huomioida ohjelmistotuotelinjan koko elinkaari. (Clements & Northrop, 2002.)

Toinen vaihe vaatimusten määrittelyssä Van der Lindenin, Schmidin ja Rommesin (2007) mukaan on vaatimusten dokumentointi. Vaatimukset tulisi dokumentoida selkeästi ja ymmärrettävästi (Van der Linden, Schmid & Rommes, 2007). Tämä voi pienentää riskiä kommunikaatio-ongelmista, joita voi esiintyä vaatimusten myöhemmässä tulkinnassa eri sidosryhmien välillä (Clements & Northrop, 2002).

Kolmas vaihe on neuvottelu, jossa vaatimuksista pyritään pääsemään yhteisymmärrykseen (Van der Linden, Schmid & Rommes, 2007). Tässä vaiheessa siis pyritään vielä varmistamaan, että kaikki sidosryhmät ymmärtävät vaatimukset samalla tavalla. Näin voidaan varmistaa, että lopulliset komponentit vastaavat kaikkien odotuksia.

Neljäs vaihe Van der Lindenin, Schmidin ja Rommesin (2007) mukaan on vaatimusten validointi ja varmistaminen. Tässä vaiheessa vaatimukset muotoillaan lopulliseen selkeään ja ymmärrettävään muotoon. On tärkeää myös varmistaa, että vaatimukset ovat järkeviä tuotteen kannalta (Clements & Northrop, 2002).

Van der Lindenin, Schmidin ja Rommesin (2007) mukaan vaatimusten määrittelyn viimeisenä vaiheena on vaatimusten hallinta. Vaatimusten hallinta vaihe jatkuu käytännössä koko ohjelmistotuotelinjan elinkaaren ajan. Vaiheen keskeisenä tehtävänä on vaatimuksien ylläpito. Tarvittaessa vaatimusten hallinnasta siirrytään takaisin aikaisempiin vaiheisiin tai luodaan kokonaan uusia vaatimuksia. (Van der Linden, Schmid & Rommes, 2007.)

Ydinkomponenttien vaatimusten määrittelyn tuloksena tulisi olla selkeästi ja ymmärrettävästi määritetyt vaatimukset ohjelmistotuotelinjan alustalle. Osa näistä vaatimuksista koskee kaikkia ohjelmistotuotelinjan tuotteita, mutta vaatimuksissa tulisi kuitenkin huomioida variaatiot tuotteiden välillä. Tämä mahdollistaa vaatimusten hyödyntämisen pohjana tuotteen vaatimuksen määrittelylle. (Van der Linden, Schmid & Rommes, 2007.)

### 4.2.3 Arkkitehtuurin määrittely

Van der Lindenin, Schmidin ja Rommesin (2007) määrittelemä kolmas alaproessi ydinkomponenttien kehittämisessä on arkkitehtuurin määrittely. Tässä alaprosessissa määritellään ohjelmistotuotelinjan ydinarkkitehtuuri. Ydinarkkitehtuuri määritellään ydinkomponenttien vaatimusten pohjalta. Ydinarkkitehtuuri toimii pohjana tuotteen arkkitehtuurin luomiselle. (Van der Linden, Schmid & Rommes, 2007.)

Ohjelmistotuotelinjan ydinarkkitehtuuri määrittelee sen, millaisia ydinohjelmistokomponentteja ohjelmistotuotelinjan ydinkomponentteihin tarvitaan (Clements & Northrop, 2002). Eli arkkitehtuuri siis määrittelee millaisia ominaisuuksia ja variaatiopisteitä ohjelmistokomponenteissa tulisi olla. Ohjelmistotuotelinjan arkkitehtuuri säilyy pitkälti samanlaisena koko ohjelmistotuotelinjan elinkaaren ajan (Clements & Northrop, 2002).

#### 4.2.4 Ydinohjelmistokomponenttien luominen tai hankinta

Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä neljäs ydinkomponenttien kehittämisprosessin alaprosessi on ydinohjelmistokomponenttien luominen tai hankinta. Tässä alaprosessissa luodaan tai hankitaan ohjelmistotuotelinjan hyödyntämät ydinohjelmistokomponentit. Näiden komponenttien suunnittelu ja luominen tapahtuvat ohjelmistotuotelinjan vaatimusten ja arkkitehtuurin pohjalta (Clements & Northrop, 2002).

Tyypillisesti organisaatiolla on neljä eri keinoa hankkia ydinohjelmistokomponentteja. Ensimmäinen keino on luoda ne itse. Luomalla komponentteja itse, organisaatiolla on täysi hallinta siitä, millainen komponentista tulee. (Van der Linden, Schmid & Rommes, 2007.) Komponentti voidaan siis luoda täysin organisaation vaatimuksien mukaisesti.

Toinen keino ydinohjelmistokomponenttien hankintaan on niiden ostaminen. Jotkin komponentit voivat olla laajuudeltaan niin suuria, että niiden luominen itse vaatisi liian paljon resursseja (Van der Linden, Schmid & Rommes, 2007). Näissä tapauksissa voidaan hyödyntää valmiita komponentteja, joita on ostettavissa markkinoilta. Tällaisia voivat olla esimerkiksi käyttöjärjestelmät tai ohjelmistokehitystyökalut (Clements & Northrop, 2002).

Van der Lindenin, Schmidin ja Rommesin (2007) esittämä kolmas keino ohjelmistoydinkomponenttien hankintaan on niiden etsiminen olemassa olevista organisaation tuotteista tai järjestelmistä. Organisaatio voi siis tarkastella esimerkiksi aikaisempia tuotteitaan ja etsiä sieltä mahdollisesti hyödyllisiä ydinkomponentteja ohjelmistotuotelinjaan. Haasteena tässä voi olla löydettyjen komponenttien muokkaaminen ohjelmistotuotelinjan vaatimuksiin tai komponenttien laatu. (Van der Linden, Schmid & Rommes, 2007.) Ennen komponentin hyväksymistä ydinkomponentteihin, se tulisikin analysoida hyvin tarkasti ja varmistaa, että se täsmää ydinkomponenttien arkkitehtuuriin ja vaatimuksiin (Clements & Northrop, 2002).

Viimeisenä ydinohjelmistokomponenttien hankintakeinona Van der Linden, Schmid ja Rommes (2007) esittävät komponentin kehittämisen ulkoistamisen. Tässä tilanteessa komponentin kehitystyö ja luominen ostetaan toiselta yritykseltä. Kehitystyön ulkoistamisessa on tärkeää, että komponentin vaatimukset on selkeästi määritellyt. Kolmannen osapuolen käsitys vaatimuksista tulisi olla täysin sama, kun organisaatiolla, joka on vaatimukset luonut. Muuten riskinä on, että tuotettu komponentti ei vastaa organisaation tarpeita. (Van der Linden, Schmid & Rommes, 2007.) Kehitystyön ulkoistamisessa tulee huomioida myös kehittäjäyrityksen luotettavuus ja tehokkuus (Clements & Northrop, 2002). Organisaation tulee olla varma, että he saavat laadukkaan ja vaatimuksien mukaisen komponentin.

#### 4.2.5 Testaaminen

Viimeisenä ydinkomponenttien kehittämisen alaprosessina Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä on testaaminen. Testaaminen on

hyvin merkittävässä osassa ydinkomponenttien kehittämistä, sillä ydinohjelmistokomponentteja hyödynnetään jokaisessa ohjelmistotuotelinjan tuotteessa, joten niiden toimivuus ja laatu on varmistettava. Usein ohjelmistotuotelinjan testaaminen on hyvin työläs ja kallis prosessi (Engström & Runeson, 2011).

Ydinohjelmistokomponenttien testaamisessa ongelmana on usein niiden sisältämät variaatiopisteet. Pienikin variaatio tietyssä ohjelmistokomponentissa lisää huomattavasti erilaisten testien määrää, joita komponentille pitäisi suorittaa. Variaatiopisteiden lisäksi ydinkomponenttien testaamisen haasteena on eri komponenttien yhdistelmien suuri määrä. (do Carmo Machado, McGregor & Santana de Almeida, 2012.) Tämän vuoksi onkin mahdotonta testata kaikkia mahdollisia tuotteita etukäteen ydinkomponenttien pohjalta (Van der Linden, Schmid & Rommes, 2007). Suurena riskinä ohjelmistotuotelinjan ydinkomponenttien testaamisessa onkin sen riittämättömyys (Clements & Northrop, 2002).

Ydinohjelmistokomponenttien testaamiseen on olemassa lukuisia erilaisia keinoja. Van der Linden, Schmid ja Rommes (2007) esittävät kolme erilaista testityyppiä, joita ohjelmistotuotelinjan ydinohjelmistokomponenttien testaamisessa voidaan hyödyntää: dynaamiset testit, regressiotestit ja hyväksymistestit.

Dynaamiset testit sisältävät yksittäisten komponenttien yksikkötestit, komponenttien väliset integraatiotestit sekä koko järjestelmän järjestelmätestit (Van der Linden, Schmid & Rommes, 2007). Yksikkötestit ovat ohjelmistokomponentille suoritettavia testejä, jotka varmistavat komponentin toiminnan. Yksikkötestejä suoritetaan tyypillisesti jo komponentin kehitysvaiheessakin. (Clements & Northrop, 2002.)

Regressiotestejä hyödynnetään varmistamaan komponentin toimivuus sen muutoksen tai kehittymisen jälkeen. (Van der Linden, Schmid & Rommes, 2007.) Regressiotesteillä siis varmistetaan, että komponentti toimii edelleen halutulla tavalla muutosten jälkeen. Regressiotestejä suoritetaan yleensä säännöllisesti laajalle joukolle komponentteja muutosten jälkeen (Clements & Northrop, 2002). Näin voidaan varmistaa, että komponentit toimivat myös kokonaisuutena yhden komponentin muutoksien jälkeen.

Hyväksymistestejä suoritetaan esimerkiksi kolmannen osapuolen tekemille ohjelmistokomponenteille, jotta voidaan varmistua, että ne täyttävät niille asetetut vaatimukset. Hyväksymistestejä voidaan myös suorittaa ohjelmistotuotelinjan koko alustalle yhdessä asiakkaan kanssa, eli tässä tapauksessa henkilöiden kanssa, jotka suorittavat tuotteen kehittämistä alustan pohjalta. (Van der Linden, Schmid & Rommes, 2007.) Hyväksymistestit eroavat muista testeistä siten, että hyväksymistesteissä testattava komponentti testataan realistisesti asiakkaan todellisessa ympäristössä (Clements & Northrop, 2002).

### 4.3 Tuotteen kehittäminen

Ohjelmistotuotelinjan tehokkaan toiminnan kannalta ydinkomponenttien kehittäminen on usein keskeisenä prosessina. Tuotteiden kehittäminen ydinkomponenttien pohjalta on kuitenkin ohjelmistotuotelinjan keskeinen lopullinen tavoite.

(Clements & Northrop, 2002.) Tuotteiden toimittaminen asiakkaalle on kuitenkin usein koko liiketoiminnan elinehto ja mahdollistaa koko organisaation toiminnan.

Ohjelmistotuotelinjan tuotteen kehitys pohjautuu aina ohjelmistotuotelinjan ydinkomponentteihin (Van der Linden, Schmid & Rommes, 2007). Jokaisella tuotteella on kuitenkin omat yksilölliset vaatimukset, joita ydinkomponentit eivät välttämättä pysty täyttämään (Clements & Northrop, 2002). Näitä tuotekohtaisia vaatimuksia on pyritty huomioimaan ydinkomponenteissa, mutta tuote harvoin kuitenkaan muodostuu käyttämällä pelkästään ohjelmistotuotelinjan ydinkomponentteja. Tässä tapauksessa tuotetta varten pitää luoda uusia tuotekohtaisia komponentteja. (Van der Linden, Schmid & Rommes, 2007.)

Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä tuotteen kehityksen prosessi on jaettu neljään alaprosessiin. Nämä prosessit ovat vaatimusten määrittely, arkkitehtuurin määrittely, tuotteen luominen ja tuotteen testaaminen. Kunkin alaprosessin pohjana toimii pitkälti ydinkomponenttien kehittämisen vastaavien alaprosessien tuotokset. Näitä joudutaan kuitenkin muokkaamaan tuotekohtaisten vaatimuksien pohjalta. (Van der Linden, Schmid & Rommes, 2007.) Tuotteen kehittämisen alaprosessit Van der Lindenin, Schmidin ja Rommesin (2007) mukaan tiivistettynä taulukossa 2.

Taulukko 2 Tuotteen kehittämisen prosessit. (Van der Linden, Schmid & Rommes, 2007)

<b>Prosessi</b>	<b>Keskeinen sisältö</b>
Vaatimusten määrittely	Määritellään tuotteen vaatimukset eli mitä toimintoja ja ominaisuuksia tuotteessa tulisi olla.
Arkkitehtuurin määrittely	Määritellään tuotteen arkkitehtuuri. Pohjana toimii ydinkomponenttien arkkitehtuuri, mutta tuotekohtaiset vaatimukset tulee huomioida.
Tuotteen luominen tai hankinta	Luodaan asiakkaan vaatimusten mukainen tuote ydinkomponenttien ja mahdollisten tuotekohtaisten komponenttien avulla. Tuotekohtaiset komponentit voidaan luoda alusta asti uutena, muokata ydinkomponenteista tai hyödyntää ydinkomponentin variaatiota.
Testaaminen	Testataan tuotetta ja varmistetaan, että se täyttää annetut vaatimukset. Testauksen painotus uusiin tuotekohtaisiin komponentteihin ja ominaisuuksiin.

### 4.3.1 Vaatimusten määrittely

Tuotteen kehittämisprosessi alkaa tuotteen vaatimuksien määrittelyllä. Tuotteen vaatimukset määrittelevät tarkalleen sen, mitä ominaisuuksia ja toimintoja tuotteessa tulisi olla. Nämä vaatimukset tulevat usein tuotteen sidosryhmiltä kuten asiakkaalta, eli tuotteen loppukäyttäjältä. (Van der Linden, Schmid & Rommes, 2007.)

Ydinkomponenttien vaatimukset toimivat pohjana tuotteen vaatimuksille, mutta ne eivät kuitenkaan aina kata kaikkia asiakkaan vaatimia ominaisuuksia (Van der Linden, Schmid & Rommes, 2007). Tämä asiakkaan ja ydinkomponenttien vaatimusten välinen eroavaisuus sisältää usein ominaisuuksia, jotka pitää toteuttaa tuotekohtaisesti. Niitä ei siis kyetä toteuttamaan ydinkomponenttien avulla. On kuitenkin myös mahdollista, että asiakkaan vaatimuksia saadaan mukautettua niin, että ydinkomponentit kattavat sen tai asiakas voi luopua vaatimuksesta (Van der Linden, Schmid & Rommes, 2007).

Tuotteen vaatimuksia määriteltäessä olisi hyvä huomioida toiminnallisten vaatimusten lisäksi myös muita tuotteen vaatimuksia (Clements & Northrop, 2002). Näitä vaatimuksia voivat olla esimerkiksi erilaiset laadulliset vaatimukset kuten luotettavuus, tehokkuus tai turvallisuus. Näiden vähäinen huomiointi on yksi merkittävä riski tuotteen vaatimusten määrittelyssä. (Clements & Northrop, 2002.)

### 4.3.2 Arkkitehtuurin määrittely

Toisena alaprozessina tuotteen kehityksen prosessissa Van der Lindenin, Schmidin ja Rommesin (2007) mukaan on tuotteen arkkitehtuurin määrittely. Tässä prosessissa määritetään tuotteen arkkitehtuuri, joka kattaa kaikki tuotteen vaatimukset. Tuotteen arkkitehtuurin pohjana toimii ydinkomponenttien arkkitehtuuri, mutta sitä tulee kuitenkin muokata tuotekohtaisten vaatimuksien pohjalta. (Van der Linden, Schmid & Rommes, 2007.)

Tuotteen arkkitehtuuria määriteltäessä, on mahdollista, että kohdataan variaatiopisteitä, jotka eivät sovellu ohjelmistotuotelinjan ydinarkkitehtuuriin (Clements & Northrop, 2002). Näissä tilanteissa asiasta tulisi ilmoittaa ydinkomponenttien kehittämisestä vastaavalle ryhmälle. Tämän palautteen pohjalta ydinarkkitehtuuria voidaan muokata tai kehittää, jotta uusi variaatiopiste voidaan huomioida tulevissa tuotteissa (Clements & Northrop, 2002).

### 4.3.3 Tuotteen luominen tai hankinta

Van der Lindenin, Schmidin ja Rommesin (2007) viitekehyksessä tuotteen kehittämisen kolmas alaprosessi on tuotteen luominen. Tuotteen luomisessa keskeistä on luoda asiakkaan vaatimuksien mukainen tuote. Tuotteen luonnissa käytetään ohjelmistotuotelinjan ydinkomponentteja, jotta tuote voidaan luoda mahdollisimman nopeasti ja kustannustehokkaasti (Van der Linden, Schmid & Rommes, 2007).

Tuotetta ei kuitenkaan usein pystytä luomaan kokonaan ohjelmistotuotelinjan ydinkomponenttien pohjalta. Tässä tilanteessa tuotetta varten pitää luoda uusia tuotekohtaisia komponentteja. Nämä tuotekohtaiset komponentit luodaan ensisijaisesti vain tiettyä tuotetta varten, mutta niistä voi myös myöhemmin muodostua ohjelmistotuotelinjan ydinkomponentteja, mikäli sille nähdään tarvetta myös tulevissa tuotteissa. (Van der Linden, Schmid & Rommes, 2007.)

Clements ja Northrop (2002) esittävät neljä eri keinoa, miten komponentteja saadaan tuotetta varten. Ensimmäinen keino on hyödyntää suoraan valmista komponenttia ydinkomponenteista. Toinen keino on hyödyntää valmista ydinkomponenttia ja implementoida siihen valmiina oleva variaatio. Kolmas keino valmiin komponentin hyödyntäminen muokkaamisen jälkeen. Viimeinen keino on uuden komponentin luominen. (Clements & Northrop, 2002.)

Tuotekohtaisten uuden komponenttien luominen on usein yksinkertaisempaa kuin uuden ydinkomponenttien luominen (Clements & Northrop, 2002). Tämä johtuu siitä, että uusi tuotekohtainen komponentti luodaan vain tätä tiettyä tuotetta varten, joten sen kehittämisessä ei ole välttämättä tarpeellista huomioida variaatiopisteitä. Ohjelmistotuotelinjan tulevaisuuden kannalta olisi kuitenkin hyödyllistä pohtia variaatiopisteitä jo tässä vaiheessa, sillä on mahdollista, että tuotekohtaisia komponentteja lisätään myöhemmin ydinkomponenteiksi (Clements & Northrop, 2002).

#### 4.3.4 Testaaminen

Viimeinen vaihe tuotteen kehittämisprosessissa on tuotteen testaaminen. Tässä vaiheessa testataan, että luotu tuote täyttää sille asetetut vaatimukset. (Van der Linden, Schmid & Rommes, 2007.) Osa tuotteesta on testattu jo ydinkomponenttien testausvaiheessa, mutta nämäkin komponentit tulee testata uudelleen. Näin voidaan varmistua, että kaikki komponentit toimivat odotetulla tavalla valmiissa tuotteessa (Van der Linden, Schmid & Rommes, 2007).

Testauksessa tulee kiinnittää erityisesti huomiota ydinkomponenttien variaatiopisteisiin ja tuotekohtaisiin uusiin komponentteihin (Van der Linden, Schmid & Rommes, 2007). Uusien komponenttien testaaminen on tärkeää, sillä näitä komponentteja ei ole aikaisemmin hyödynnetty ohjelmistotuotelinjan tuotteissa. Testauksella varmistetaan, että ydinkomponentit ja tuotekohtaiset komponentit toimivat yhdessä odotetulla tavalla. (Clements & Northrop, 2002.)

## 4.4 Johtaminen ja hallinta

Kolmas keskeinen prosessi ohjelmistotuotelinjan toiminnassa on johtaminen ja hallinta (Clements & Northrop, 2002). Johtamisessa ja hallinnassa keskeistä on, että johtohenkilöstö on sitoutunut ohjelmistotuotelinjan toimintaan niin teknisellä kuin organisaatiosollakin. Johtamisen ja hallinnan tehtävänä teknisellä tasolla on valvoa ja hallita ydinkomponenttien sekä tuotteiden kehittämistä. Tässä erityisesti tulee kiinnittää huomiota siihen, että ydinkomponenttien ja tuotteen

kehittämisen yksiköt toimivat oikeiden aktiviteettien parissa ja noudattavat ohjelmistotuotelinjalle määritettyjä prosesseja. (Clements & Northrop, 2002.)

Organisaatiotasolla johdon tulee pitää huoli, että organisaatorakenne on ohjelmistotuotelinjalle soveltuva (Clements & Northrop, 2002). Keskeisenä ominaisuutena ohjelmistotuotelinjaan keskittyvällä organisaatiolla on selkeä jako ydinkomponentteja kehittävän yksikön ja tuotteita kehittävän yksikön välillä (Van der Linden, Schmid & Rommes, 2007). Organisaation johdon on myös tärkeää pitää huolta, että kaikilla ohjelmistotuotelinjan toimintaan liittyvillä yksiköillä on riittävästi resursseja käytössään (Clements & Northrop, 2002). Tämä takaa sen, että esimerkiksi ydinkomponenttien ja tuotteen kehittämisen yksiköt kykenevät suorittamaan omia tehtäviään tehokkaasti.

## 5 HAASTATTELUJEN ANALYYSI

Tässä luvussa analysoidaan suoritettut haastattelut. Haastateltavaksi valittiin kolme henkilöä, jotka toimivat keskeisissä rooleissa ohjelmistotuotelinjan toiminnan ja kehittämisen kannalta. Haastatteluilla pyrittiin selvittämään organisaation nykyisiä toimintatapoja ohjelmistotuotelinjan osalta. Aluksi käydään läpi mitä tavoitteita haastateltavat kertoivat ohjelmistotuotelinjalle. Seuraavaksi käsitellään organisaation nykyisestä toiminnasta tunnistettuja haasteita. Lopuksi esitellään nykyisiä käytössä olevia ja toivottuja prosesseja sekä käytäntöjä organisaation ohjelmistotuotelinjan toiminnasta.

### 5.1 Organisaation ohjelmistotuotelinjan tavoitteet

Haastatteluista nousi esille paljon yhtäläisyyksiä eri haastateltavien kesken. Yhtenä keskeisenä yhdistävänä tekijänä haastatteluissa oli selkeä tavoite ohjelmistotuotelinjalle. Kaikilla haastateltavilla oli oma selkeä näkemys siitä millainen ohjelmistotuotelinjan tulisi olla ja miten sen tulisi toimia.

Kaikki haastateltavat ilmaisivat, että heidän mielestään ohjelmistotuotelinjaa tulisi ohjata sellaiseen suuntaan, että se pystyisi tarjoamaan asiakkaille mahdollisimman paljon vakioratkaisuja. Vakioratkaisuilla he tarkoittavat, sitä, että asiakkaille pyritään myymään valmiita kokonaisuuksia, jotka pystytään toteuttamaan mahdollisimman hyvin ohjelmistotuotelinjan ydinkomponenttien pohjalta.

”Minun mielestä siihen meidän pitäisi pyrkiä, että myydään sitä mitä meillä jo on.” (Henkilö 3, haastattelu, 15.5.2023)

Vakioratkaisujen hyödyntäminen muuttaisi ohjelmistotuotelinjan toimintaa siten, että yksittäistä tuotetta varten tehtyjen kustomoitujen komponenttien tai ominaisuuksien toteuttamista pyrittäisiin minimoimaan. Mahdolliset tuotekohtaiset ominaisuudet pyrittäisiin huomioimaan mahdollisimman hyvin variatiopisteiden avulla.



Vakioratkaisujen avulla tuotteiden valmistus- ja toimitusaikoja kyettäisiin lyhentämään huomattavasti, kun uusien komponenttien kehittämiseen ei tarvitsi käyttää niin paljon resursseja. Tämä kuitenkin vaatii paljon investointeja ydinkomponenttien kehittämiseen, jotta mahdollisimman suuri osa mahdollisista ominaisuuksista saadaan toteutettua ydinkomponentteihin. Yksi haastateltavista kommentoi, että ihanteellisessa tilanteessa kaikki tuotteet kyettäisiin luomaan pelkästään ydinkomponenttien pohjalta pelkän konfiguroinnin avulla. Samalla hän kuitenkin totesi, että tämä ei kuitenkaan ole ohjelmistotuotelinjalle kovinkaan realistinen tavoite.

Yhtenä selkeänä tavoitteena ohjelmistotuotelinjalle mainittiin myös yksinkertaisesti kyky valmistaa tuotteita sen pohjalta ja kyky ylläpitää ja kehittää ohjelmistotuotelinjaa. Lisäksi ohjelmistotuotelinjalta toivottiin selkeämpää versionhallintaa ja kattavampaa dokumentointia toiminnoista sekä variaatiopisteistä kuin nykyisessä toimintatavassa.

”Modulaarinen platformi, joka tarjoaa sellaisen softa arkkitehtuurin ja rakenteen, että projekteja on mahdollisuus tehdä ja teknologiaosaston on mahdollista ylläpitää ja laajentaa platformia.” (Henkilö 3, haastattelu, 15.5.2023)

”Platformin softa-arkkitehtuuri ja variaatiopisteet on implementoitu ja dokumentoitu sillä tavalla, että projekteja on mahdollista tehdä.” (Henkilö 3, haastattelu, 15.5.2023)

Kirjallisuuteen peilattuna haastatteluissa esille nousseet tavoitteet vastasivat hyvin kirjallisuudessakin esitettyjä tavoitteita ohjelmistotuotelinjalle. Kirjallisuudessa ohjelmistotuotelinjan tavoitteet keskittyivät juuri organisaation tuotteiden valmistamisen kyvyn kehittämiseen sekä tuotteiden laadun parantamiseen ja kustannuksien pienentämiseen.

## 5.2 Tunnistettuja haasteita

Haastatteluissa ilmeni myös paljon haasteita, joita haastateltavat ovat kokeneet ohjelmistotuotelinjan nykyisessä toiminnassa. Yhtenä yleisenä haasteena oli selkeän toimintamallin puute. Yhden haastateltavan mukaan monet tämän hetkistä toimintatavoista perustuvat vain eri henkilöiden omaan kokemukseen asiasta. Mitään selkeää määriteltyä toimintatapaa ei siis välttämättä ole kaikille prosesseille. Toisen haastateltavan mukaan nykyisen toiminnan haasteena on myös se, että eri henkilöillä on erilaisia mielipiteitä siitä, miten asioita tulisi tehdä. Selkeästä yhteisestä toimintamallista voisi siis olla hyötyä organisaation toiminnassa.

”Selkeitä haasteita ymmärtää mitä meiltä halutaan... Ei osata päättää moduulijakoa, selkeästi erilaisia näkemyksiä ihmisillä.” (Henkilö 1, haastattelu, 17.5.2023)

Toinen esille noussut haaste on vaatimusten määrittely. Tämä oli selkeästi suurin haaste ohjelmistotuotelinjan toiminnassa tällä hetkellä. Kaikki haastateltavat mainitsivat tämän selkeäksi haasteeksi organisaatiossa. Tämä on selkeästi haaste organisaatiolle, koska organisaation tämän hetkinen toimintatapa on vielä hyvin projektipainotteinen. Organisaatiossa ei siis vielä kunnolla hyödynnetä ohjelmistotuotelinjaa.

”Tällä hetkellä vaatimukset tulevat softakavereiden kokemuksesta, eli ne ovat itse keksittyjä. Todettu, että tällaiset ovat aika monessa projektissa toimineet ... Pitäisi pystyä nämä naulitsemaan, nyt liikaa tilanteita, että projektissa todetaan, että nämä eivät toimikkaan.” (Henkilö 2, haastattelu, 25.5.2023)

Projektipainotteisuus näkyy organisaatiossa siten, että ohjelmistotuotteita valmistetaan aina tiettyä asiakastilasta varten. Näissä projekteissa ohjelmiston vaatimukset tulevat pitkälti suoraan asiakkaalta, joten vaatimusten määrittely on suhteellisen suoraviivainen prosessi. Uuden ohjelmistoalustan ohjelmistotuotelinjan tapauksessa vaatimukset eivät tulekaan suoraan ulkopuoliselta asiakkaalta, vaan ohjelmistotuotelinjan ydinkomponenttien vaatimukset tulisi määrittellä organisaation sisäisten sidosryhmien kesken. Sisäisessäkin vaatimusten määrittelyssä pohjana toimivat varmasti mahdollisten asiakkaiden tarpeet, mutta päätös tarvittavista vaatimuksista tulee tehdä organisaation sisällä.

”Organisaation pitäisi pystyä itse luomaan tietynlaiset raamit toiminnalle. Ennen se on tullut suoraan asiakkaalta.” (Henkilö 3, haastattelu, 15.5.2023)

Tämä organisaation sisäinen vaatimusten määrittely ohjelmistotuotelinjan ydinkomponenteille on osoittautunut hyvin haastavaksi. Yhden haastateltavan mukaan haastavaa on erityisesti se, että ohjelmistotuotelinjalle saattaa tulla tietty toiminnallinen tarve, mutta sen tarkempi vaatimusten määrittely esimerkiksi teknisen toteutuksen kannalta on hyvin vajaa. Näissä tapauksissa yksittäisellä ohjelmistonkehittäjällä on todella paljon vastuuta toiminnallisuuden luomisessa. Tämä ei ole ohjelmistotuotelinjan kannalta järkevää, vaan vaatimusten määrittelyn tulisi olla tarkempi jo ennen komponenttien luomisen aloittamista. Näin voidaan varmistua, että komponentti varmasti täyttää siltä odotetut vaatimukset ja kaikilla sidosryhmillä on yhtenevä käsitys niistä.

”Vaatimukset ei aina loppuun asti mietittyjä, usein mietti, että mennäänkö tämän kanssa metsään.” (Henkilö 1, haastattelu, 17.5.2023)

Tämän hetkessä projektipainotteisessa toiminnassa on ilmennyt myös haasteita palautteen antamisessa. Yksi haastateltavista nosti esille ongelman, että tuotteen luomisesta vastaava osasto ei välttämättä anna palautetta alustasta esimerkiksi sen ominaisuuksiin liittyen. Haastateltava toivoi, että palautetta annettaisiin esimerkiksi tarpeellisista ominaisuuksista joita tulisi lisätä ohjelmistotuotelinjan ydinkomponentteihin. Tällä hetkellä voi olla tilanne, että monissa projekteissa saatetaan luoda jokin sama ominaisuus useaan kertaan sen sijasta, että

tuo ominaisuus olisi lisätty ydinkomponentteihin. Uusissa projekteissa saatetaan siis tehdä ylimääräistä työtä.

”Alusta ei kehity ... Joka projekti tekee samat asiat uudestaan eri tavalla ... Täytyisi päästä koko ajan eteenpäin.” (Henkilö 1, haastattelu, 17.5.2023)

Yhtenä haasteena haastatteluissa nousi myös esille ohjelmistotuotelinjan integroiminen yhteen muiden ohjelmistojen kanssa. Organisaation sisällä on useita eri ohjelmistoja, joiden kanssa ohjelmistotuotelinjan tulee toimia. Kun ohjelmistotuotelinjan ydinkomponentteja kehitetään, onkin tärkeää huomioida toisten ohjelmistojen rajapinnat ja varmistaa, että uudet ominaisuudet ovat yhteensopivia näiden kanssa. Tämä tuo ydinkomponenttien kehittämiseen lisää haastetta. Ohjelmistojen yhteensopivuus tuo vaikutuksia tietysti myös toiseen suuntaan. Mikäli muiden ohjelmistojen toimintaan tai rajapintaan tulee muutoksia, voivat ne vaatia myös ohjelmistotuotelinjan ydinkomponenttien muokkaamista.

### 5.3 Toimintatapoja

Tutkittava ohjelmistotuotelinja on organisaatiossa suhteellisen uusi. Ohjelmistotuotelinjan pohjalta ei ole vielä rakennettu monia varsinaisia asiakkaille toimitettavia tuotteita, mutta ydinkomponenttien eli ohjelmistotuotelinjan alustan kehittäminen on kuitenkin jo käynnissä. Haastatteluissa esiin nousseet toimintatavat ja käytännöt liittyivätkin enimmäkseen ohjelmistotuotelinjan ydinkomponenttien kehittämiseen. Ohjelmistotuotelinjan tuotteiden kehittämistä ei ole vielä varsinaisesti toteutettu, joten tämän osalta, toimintatavat ja menetelmät ovat lähinnä haastateltavien mielipiteitä siitä, miten tuotteen kehittämisen tulisi toimia.

”Ihmisillä erilaisia ajatuksia ohjelmistotuotelinjan toiminnasta. Sitä ei oikein ole paperilla mietitty.” (Henkilö 1, haastattelu, 17.5.2023)

Haastatteluissa nousi esille, että ohjelmistotuotelinjaa pyritään kehittämään projektina SCRUM-viitekehyksen avulla. SCRUM-viitekehyyksessä työtä tehdään tyypillisesti kolmen viikon mittaisissa sprinteissä. Sprintille määritellään aina tietyt tavoitteet ja sprintin lopussa käydään läpi, miten sprintin tavoitteet ovat täyttyneet. Organisaatiossa SCRUM on jaettu vielä neljännes vuoden mittaisiin iteraatioihin, joissa kussakin toteutetaan kolme sprinttiä. SCRUM-viitekehyksen avulla kehitystyöstä saadaan hyvin organisoitua ja työn tavoitteet pysyvät selkeinä.

Haastattelujen perusteella yhtenä keskeisenä osana ohjelmistotuotelinjan ydinkomponenttien kehittämistä on vaatimusten määrittely. Organisaatiossa pyritään toteuttamaan jatkuvaa vaatimusten määrittelyä, jolla kartoitetaan mahdollisia uusia ominaisuuksia ydinkomponentteihin. Kuten aiemmin todettu niin vaatimusten määrittely on kuitenkin todettu hyvin haastavaksi ilman selkeitä toiveita asiakkaalta.

”Requirement management on jatkuva prosessi, joka sisältää sisäiset sidosryhmät. Tarkoituksena tuottaa lista vaatimuksia... Toiminnalliset vaatimukset ja siihen liittyvät ei-toiminnalliset vaatimukset.” (Henkilö 3, haastattelu, 15.5.2023)

Ydinkomponenttien vaatimusten määrittelyssä organisaatiossa hyödynnetään esimerkiksi myyntiosaston kokemuksia ja käsityksiä asiakkaiden mahdollisista tarpeista. Vaatimusten määrittelyä pyritään toteuttamaan usean eri osaston yhteistyöllä. Myyntiosaston lisäksi apuna on esimerkiksi solution management-tiimi sekä ohjelmistokehitystiimi. Vaatimusten määrittelyn tavoitteena on tuottaa sekä selkeät toiminnalliset ja ei-toiminnalliset vaatimukset. Vaatimuksista tulisi siis käydä ilmi sen toiminta ja tekninen toteutus.

”Myynti ja solution ovat yhteistyössä, sieltä tulee inputtina, että mitä asiakkaalle myydään ja mitä he vaativat. Eli heillä on ratkaisu ja me toteutamme sitä. Välillä tämä kuitenkin on epäselvää.” (Henkilö 1, haastattelu, 17.5.2023)

Organisaatiossa vaatimusten määrittelyn jälkeen ohjelmistotuotelinjan tuoteomistaja esittää toiveen jonkin ominaisuuden toteuttamisesta. Tuoteomistaja esittelee tässä vaiheessa siis haluttavan ominaisuuden ja siihen liittyvät dokumentaatiot. Tämä toive uudesta ominaisuudesta esitellään kehitystiimille, joka olisi siis vastuussa ominaisuuden toteuttamisesta. Haastattelujen perusteella kehitystiimi usein tietyllä tapaa haastaa tuoteomistajaa ominaisuudesta ja sen toiminnasta ja toteutuksesta käydään keskustelua tarkemmin.

Seuraavassa vaiheessa kehitystiimi pitää oman sisäisen palaverin. Tässä palaverissa käydään läpi ominaisuutta ja pohditaan sen teknistä toteutusta. Kehitystiimin sisäisen palaverin jälkeen he esittävät tuoteomistajalle tarjouksen ominaisuudesta. Tässä kohtaa tuoteomistajalle esitellään valmis ehdotus toteutettavasta ominaisuudesta ja sen teknisestä toteutuksesta. Ominaisuus ei kuitenkaan välttämättä ole täysin samanlainen kuin mitä tuoteomistaja on alun perin esitellyt kehitystiimille. Esimerkiksi tekniset rajoitteet voivat muokata ominaisuutta toisenlaiseksi.

”Tuoteomistaja tuo kehitystiimille vaatimuksen. Esittelee siihen liittyvät asiat ja siihen liittyvän dokumentaation. Kehitystiimillä on aikaa haastaa tuoteomistajaa ja keskustella uudesta ominaisuudesta. Tämän jälkeen kehitystiimi vetäytyy sisäiseen suunnitteluun, jonka jälkeen he tulevat teknisen suunnitelman kanssa. Tämän jälkeen kehitystiimi tarjoaa suunnitelmaa tuoteomistajalle.” (Henkilö 3, haastattelu, 15.5.2023)

Kun tarjous toteutettavasta ominaisuudesta on esitelty tuoteomistajalle, hän käy tarjouksen läpi ja tekee päätöksen siitä, että hyväksytäänkö vai hylätäänkö tarjous. Mikäli tarjous hyväksytään, alkaa kehitystiimi toteuttaa ominaisuutta ohjelmistotuotelinjan ydinkomponentteihin. Jos tuoteomistaja ei hyväksy tarjousta, niin sitä ei aleta toteuttaa ja ominaisuudesta luovutaan ainakin sellaisenaan. On kuitenkin mahdollista, että ominaisuutta pohditaan uudestaan vaatimusten määrittelyssä ja se tuodaan tarjolle uudestaan muokattuna. Vaatimusten

määrittely on siis jatkuva prosessi, joka on käynnissä jatkuvasti. Ohjelmistotuotelinjan mahdollisia uusia ominaisuuksia pyritään siis löytämään jatkuvasti, jotta ohjelmistotuotelinja vastaisi asiakkaiden tarpeita mahdollisimman hyvin.

Haastatteluissa nousi esille myös selkeä linja ohjelmistotuotelinjan ydinkomponenttien luomisesta tai hankinnasta. Organisaation tavoitteena on toteuttaa ainakin kaikki keskeisimmät ominaisuudet itse. Organisaatio pyrkii siis hyödyntämään omaa ydinosaa ydinkomponenttien toteuttamisessa. Keskeisimpien osien luomisen ulkoistamista haastateltavat eivät kokeneet järkevänä, koska organisaation oma kehitystiimi kykenee tuottamaan ne tehokkaasti. Haastateltavat kuitenkin totesivat, että laajempien, ei ydinosaan liittyvien komponenttien tai toiminnallisuuksien luomista voitaisiin ulkoistaa tai komponentit voitaisiin ostaa suoraan valmiina. Tämä päätös kuitenkin arvioitaisiin aina kustannusten ja resurssien pohjalta. Haastateltavat myös mainitsivat, että komponenttien luomisessa pyritään lähtökohtaisesti hyödyntämään jo olemassa olevia komponentteja, mutta usein komponentteja joudutaan luomaan alusta asti uutena.

”Tulevaisuudessa voisi tutkia enemmän, mikä on meidän ydinjuttu. Mitä teemme itse ja mitä voitaisiin ostaa.” (Henkilö 1, haastattelu, 17.5.2023)

Haastatteluissa nousi esille myös ydinkomponenttien suunnittelussa ja luomisessa tarve dokumentoida selkeästi mahdolliset variaatiopisteet. Variaatiopisteitä pyritään tunnistamaan heti kun mahdollista uutta komponenttia tai ominaisuutta aletaan suunnitella ohjelmistotuotelinjan ydinkomponentteihin. Yhden haastateltavan mukaan ihanteellista olisi, että komponentin eri variaatiot olisivat jo valmiiksi toteutettuna ja tuotteen kehittämisessä oikea variaatio saataisiin valittua vain konfiguroimalla. Tällä hetkellä kuitenkin variaatioiden toteutus jää usein tuotteen kehittäjien vastuulle.

Ydinkomponenttien kehittämisessä pyritään myös huomioimaan muiden ohjelmistojen rajapinnat. Haastatteluissa nousi esille se, että ohjelmistotuotelinja toimii yhdessä useiden eri ohjelmistojen kanssa, joten ydinkomponenttien kehittämisessä tulee varmistaa, että komponentit ovat yhteensopivia näiden rajapintojen kanssa. Muiden ohjelmistojen rajapinnat tulee huomioida myös tuotteen kehittämisen prosessissa, sillä tuotteeseen toteutettavat ominaisuudet saattavat vaatia myös yhteistoimintaa muiden ohjelmistojen kanssa.

Variaatiopisteet sekä eriohjelmistojen yhteen toimivuus on pitkälti riippuvaisia selkeästä dokumentoinnista. Kun variaatiopisteet sekä muiden ohjelmistojen rajapinnat ovat selkeästi dokumentoitu, voidaan ne huomioida tehokkaasti niin ydinkomponenttien kuin tuotteenkin kehittämisessä. Haastattelujen perusteella organisaatiossa pyritään panostamaan selkeään dokumentointiin, mutta kehitettävää on vielä paljon.

”Yritetään dokumentoida selkeästi enemmän, kuin aikaisemmin. Pyritään siihen, että siellä oikeasti olisi jotain hyödyllistä.” (Henkilö 1, haastattelu, 17.5.2023)

Organisaation ohjelmistotuotelinjan pohjalta ei ole kehitetty vielä monia tuotteita, joten organisaatiossa ei ole vielä paljoa vakiintuneita käytäntöjä tämän ohjelmistotuotelinjan tuotteiden kehittämiseen. Haastatteluissa nousi esille kuitenkin paljon erilaisia mielipiteitä siitä, miten tuotteiden kehittämisen tulisi toimia. Nämä mielipiteet pohjautuvat organisaation kokemukseen tuotteiden kehittämisestä projektipohjaisessa toiminnassa. Näitä käytäntöjä projektipohjaisesta tuotteen kehittämisestä voidaan kuitenkin hyödyntää myös tämän uuden ohjelmistotuotelinjan toiminnassa.

Haastattelujen perusteella tuotteen kehittäminen alkaa siitä, kun asiakkaalle myydään tuote. Tässä vaiheessa myydystä tuotteesta muodostetaan dokumentti, joka kertoo tuotteen toiminnalliset vaatimukset. Toiminnallisten vaatimusten pohjalta suoritetaan tarkempaa vaatimusten määrittelyä. Vaatimusten määrittelyn pohjalta pyritään valitsemaan ohjelmistotuotelinjan ydinkomponenteista tuotteeseen tarvittavat komponentit. Ydinkomponenteista valitaan tuotteeseen siis vain tarvittavat komponentit eli koko ohjelmistotuotelinjan alustaa ei hyödynnetä joka tuotteeseen.

”Tuotteen vaatimusten määrittelyssä lähtökohtana on asiakkaan kanssa laadittu functional specification eli toimintakuvaus, joka kuvaa mitä järjestelmän tulisi tehdä.” (Henkilö 2, haastattelu, 25.5.2023)

Haastateltavien mukaan vaatimusten määrittelyn aikana vaatimuksista tunnistetaan myös ne kohdat, joita ei ole toteutettu ohjelmistotuotelinjan ydinkomponentteihin. Nämä ovat siis ominaisuuksia, jotka pitää toteuttaa kyseistä tuotetta varten. Tuotetta varten luotavat ominaisuudet voivat olla kokonaan uusia ja ne luodaan alusta asti uutena tai ominaisuus voidaan toteuttaa uutena variaationa ydinkomponentin variaatiopisteestä. Tällä hetkellä uusien ominaisuuksien sekä ydinkomponenttien tunnistaminen pohjautuu hyvin pitkälti työntekijöiden kokemukseen. Mitään selkeää prosessia komponenttien tunnistamiselle ei siis ole organisaatiossa.

”Peilataan asiakasvaatimuksia siihen, mitä nykyiset komponentit tekevät. Se on osa tuoteomistajan ammattitaitoa katsoa, mitkä komponentit soveltuvat tuotteeseen.” (Henkilö 2, haastattelu, 25.5.2023)

Tuotekohtaisten ominaisuuksien luomiseen ei myöskään ole selkeää prosessia tällä hetkellä. Haastateltavien mukaan uutta ominaisuutta luotaessa, pyrittäisiin hyödyntämään jo olemassa olevia komponentteja, jos näitä pystyisi hyödyntämään uuden komponentin pohjana. Usein kuitenkin komponentit luodaan vain alusta alkaen uutena. Mikäli uusi ominaisuus ei liity organisaation ydinosaamiseen, olisi myös mahdollista, että komponentti ostettaisiin valmiina tai sen luominen ulkoistetaan. Tällä hetkellä kuitenkin uusia komponentteja luodaan lähinnä organisaation sisällä.

Uuden tuotekohtaisen komponentin vaatimusten määrittelyssä mukaan otetaan myös muita sidosryhmiä organisaation sisältä. Haastattelujen mukaan esimerkiksi ohjelmistotuotelinjan tuoteomistajan sekä myyntiosaston tulisi osallistua uuden ominaisuuden arviointiin. Ominaisuutta arvioidaan erityisesti sen

osalta, että olisiko sille tarvetta myös tulevissa tuotteissa. Tällä eri sidosryhmien yhteistyöllä pyritään tunnistamaan ominaisuuksia, jotka voitaisiin lisätä ydinkomponentteihin.

”Se on tasapainoilua sen välillä, mikä on osa ydintä ja mikä ei.” (Henkilö 1, haastattelu, 17.5.2023)

”Yhteistyö on avainsana. Mietitään yhdessä, miten komponentit kannattaa tehdä.” (Henkilö 2, haastattelu, 25.5.2023)

Haastattelujen perusteella voidaan todeta, että organisaatiossa ollaan mietitty ydinkomponenttien kehittämistä huomattavasti enemmän kuin tuotteen kehittämistä. Ydinkomponenttien kehittämiseen on todennäköisesti panostettu enemmän, koska ohjelmistotuotelinja on uusi, eikä tuotteita ole vielä juurikaan valmistettu sen pohjalta. Tuotteiden kehittämisen osalta haastateltavilla oli kuitenkin paljon erilaisia mielipiteitä siitä, miten tuotteita tulisi kehittää ohjelmistotuotelinjan pohjalta.

## 6 TOIMINTAMALLIN SUUNNITTELU JA LUOMINEN

Tässä luvussa käydään läpi tutkimuksen tuloksena olevan toimintamallin luomista. Luvun tavoitteena on siis muodostaa toimintamalli ohjelmistotuotelinjalle. Toimintamalli luodaan kirjallisuuden ja haastattelujen pohjalta. Aluksi käydään läpi toimintamallin tavoitteet. Tämän jälkeen käydään läpi toimintamallin luomisen prosessi ja sen eri iteraatiot. Ensimmäisessä iteraatiossa toimintamallille luodaan pohja kirjallisuudesta. Toisessa iteraatiossa toimintamallin eri vaiheita lisätään haastattelujen pohjalta. Viimeisessä iteraatiossa toimintamallia täydennetään vielä kirjallisuuden avulla. Lopuksi esitellään lopputuloksena syntynyt toimintamalli.

### 6.1 Toimintamallin tavoitteet

Tässä tutkimuksessa pyritään luomaan toimintamalli organisaation uudelle ohjelmistotuotelinjalle. Organisaation ohjelmistotuotelinjalla ei tällä hetkellä ole selkeää yleisesti hyväksyttyä toimintamallia, jonka pohjalta ohjelmistotuotelinjaa kehitetään. Ohjelmistotuotelinjan eri sidosryhmillä on vain omia mielipiteitä ja toimintatapoja, joiden pohjalta ohjelmistotuotelinjaa on kehitetty. Selkeän toimintamallin avulla organisaatio kykenee tehostamaan omaa toimintaansa ohjelmistotuotelinjan osalta, sekä kykenee hyödyntämään ohjelmistotuotelinjaa tehokkaammin tuotteiden valmistamiseen.

Toimintamalli pyrkii kuvaamaan ohjelmistotuotelinjaan liittyviä prosesseja ja keskittyy erityisesti ohjelmistotuotelinjan tekniseen puoleen eli ydinkomponenttien ja tuotteen kehittämiseen. Toimintamallin luomisessa hyödynnetään kirjallisuuskatsauksella hankittua tietopohjaa, sekä puolistrukturoiduilla haastatteluilla kerättyä tietoa organisaation nykyisistä toimintatavoista. Näiden pohjalta pyritään luomaan selkeä toimintamalli organisaation ohjelmistotuotelinjalle.



## 6.2 Toimintamallin luominen

Toimintamalli luodaan kolmessa iteraatiossa. Toimintamallin luominen aloitetaan tarkastelemalla kirjallisuuden tarjoamaa tietopohjaa. Ensimmäisessä iteraatiossa pyritään luomaan toimintamallille alustava rakenne tietopohjan avulla. Teoriapohjasta ei hyödynnetä tarkempia tietoja ja prosesseja vielä ensimmäisessä iteraatiossa, vaan tarkoituksena saada toimintamallille jonkinlainen kehys, jota pyritään täyttämään myöhemmissä iteraatioissa.

Toisessa iteraatiossa toimintamallin alustavaa kehystä aletaan täydentää haastatteluista saadulla tiedolla. Haastattelujen avulla pyrittiin saamaan esille organisaatiossa jo käytössä olevia prosesseja ja totuttuja työtapoja. Nämä toimintatavat eivät kuitenkaan ole organisaation virallisia toimintatapoja vaan lähinnä haastateltavien omia käsityksiä siitä, miten ohjelmistotuotelinjan kanssa tulisi toimia. Näitä hyödyntämällä toimintamalli mukailee organisaation valmiita ja totuttuja toimintatapoja.

Kolmannessa iteraatiossa toimintamallia täydennetään ja muokataan teoriapohjan avulla. Tässä iteraatiossa toimintamallia peilataan olemassa olevaan teoriaan ja kirjallisuuteen. Toimintamallista pyritään löytämään kohtia, joita voidaan parannella ja kehittää teorian pohjalta.

Kolmannen iteraation jälkeen lopputuloksena on toimintamalli organisaation ohjelmistotuotelinjan hallinnasta. Toimintamallin pitäisi tässä vaiheessa olla käyttökelpoinen ja selkeä. Toimintamallia kuitenkin arvioidaan vielä tutkimuksen seitsemännessä luvussa.

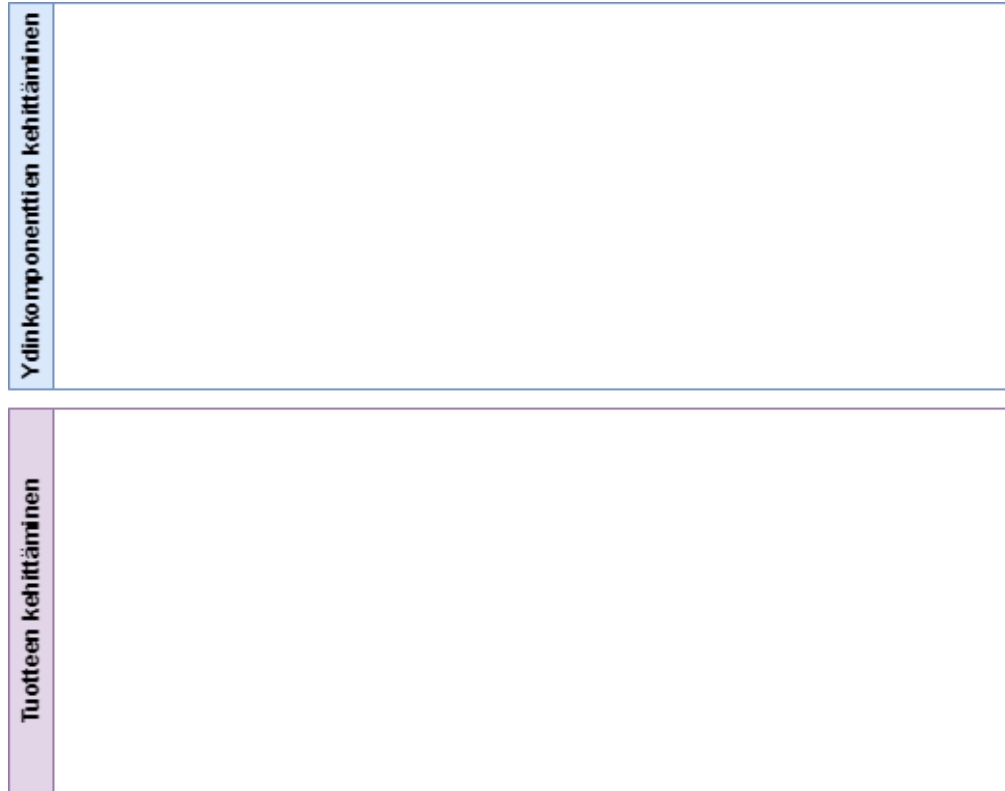
### 6.2.1 Ensimmäinen iteraatio kirjallisuuden pohjalta

Ensimmäisessä iteraatiossa pyritään luomaan toimintamallille alustava rakenne ja pohja, jota täydennetään myöhemmissä iteraatioissa. Selkeä pohja toimintamallille helpottaa sen muodostamista toisessa iteraatiossa, jossa toimintamallia täydennetään haastattelujen avulla. Pohjan luomisessa hyödynnetään kirjallisuuskatsauksella hankittua tietoa ohjelmistotuotelinjoista ja niiden prosesseista.

Kirjallisuuskatsauksessa tarkastelluissa aineistoissa nousi selkeästi esille samankaltaisia teemoja ohjelmistotuotelinjan prosesseissa. Monet aineistot nimeävät ohjelmistotuotelinjan keskeisiksi prosesseiksi ydinkomponenttien kehittämisen ja tuotteen kehittämisen. Esimerkiksi Clements ja Northrop (2002) ja Van der Linden, Schmid ja Rommes (2007) esittelevät teoksissaan ohjelmistotuotelinjoihin liittyvät viitekehukset, joissa esiintyvät ydinkomponenttien kehittämisen ja tuotteen kehittämisen prosessit.

Kirjallisuuskatsauksen pohjalta voidaan siis todeta, että ohjelmistotuotelinjan keskeisiä prosesseja ovat ydinkomponenttien kehittäminen ja tuotteen kehittäminen. Tämä jako on ohjelmistotuotelinjan toiminnan kannalta looginen, koska ohjelmistotuotelinjan toiminnassa olennaista on juurikin ydinkomponenttien luominen ja niiden pohjalta tuotteiden valmistaminen. Ohjelmistotuotelinjan keskeinen tavoite on kuitenkin kyky valmistaa kohdemarkkinoille sopivia tuot-

teita. Kirjallisuuskatsauksen pohjalta luotavan toimintamallin alustavaksi pohjaksi ja kehykseksi valitaan nämä prosessit. Toimintamallin ensimmäinen versio kuvattuna kuviossa 7.



KUVIO 7 Toimintamalli ensimmäisen iteraation jälkeen.

## 6.2.2 Toinen iteraatio haastattelujen analysoinnista

Toimintamallin kehittämisen toisessa iteraatiossa analysoidaan haastatteluista saatua tietoa ja täydennetään toimintamallia sen avulla. Tutkimuksessa suoritettiin kolme haastattelua. Haastateltavat henkilöt ovat keskeisissä rooleissa ohjelmistotuotelinjan toiminnan ja kehittämisen kannalta.

Haastatteluilla pyrittiin selvittämään heidän mielipiteitään ja käytäntöjään siitä, miten ohjelmistotuotelinjan tulisi toimia ja miten sitä tulisi kehittää. Haastattelujen avulla pyrittiin myös kartoittamaan organisaation jo käytössä olevia virallisia tai ei virallisia toimintatapoja ohjelmistotuotelinjaan liittyen ja mahdollisia ongelmakohtia niissä. Tutkimuksen kohteena oleva ohjelmistotuotelinja on kuitenkin suhteellisen uusi, joten valmiita toimintatapoja ei välttämättä vielä löydy ohjelmistotuotelinjan kaikkiin eri prosesseihin.

Haastattelujen perusteella organisaatiossa on useita eri käytäntöjä ja prosesseja ohjelmistotuotelinjan ydinkomponenttien kehittämiseen. Yhtenä selkeänä prosessina on jatkuva vaatimusten määrittely. Jatkuvan vaatimusten määrittelyn tuloksena muodostuu vaatimukset joiden perusteella ohjelmistotuotelinjan ydinkomponentteja tulisi kehittää. Tämä jatkuva vaatimusten määrittely on

siis tällä hetkellä suuressa roolissa ohjelmistotuotelinjan kehittämisesä. Vaatimusten määrittely siis lisätään luotavaan toimintamalliin ydinkomponenttien kehittämiseen.

Haastattelujen analyysissa selvisi, että vaatimusten määrittelyn jälkeiseen ydinkomponenttien suunnitteluun on olemassa selkeä toimintatapa. Vaatimusten määrittelyn jälkeen ohjelmistotuotelinjan tuoteomistaja tarkastelee vaatimuksia ja esittää niiden perusteella jotain ominaisuutta lisättäväksi ohjelmistotuotelinjan ydinkomponentteihin. Tämän jälkeen tuoteomistajan näkemystä ominaisuudesta haastetaan kehitystiimin puolelta. Kun ominaisuudesta on päästy yhteisymmärrykseen, kehitystiimi pitää oman sisäisen palaverin ominaisuuden teknisestä toteuttamisesta. Palaverissa kehitystiimi pohtii ominaisuuden tarkempaa toteutusta ja sitä mitä ominaisuudesta ylipäätään kyetään toteuttamaan. Sisäisen palaverin jälkeen kehitystiimi antaa oman esityksensä toteutettavasta ominaisuudesta ja tuoteomistaja päättää tämän pohjalta toteutetaanko kyseinen ominaisuus ydinkomponentteihin vai ei.

Organisaatiossa on selvästi pohdittu ydinkomponenttien kehittämistä ja siihen liittyviä prosesseja. Näitä valmiita toimintatapoja voidaan hyvin hyödyntää luotavassa toimintamallissa, jolloin organisaation omat käytännöt ja sidosryhmien mielipiteet näkyvät toimintamallissa.

Haastatteluissa nousi esille myös useita erilaisia mahdollisuuksia, miten uusia komponentteja luodaan tai hankitaan ydinkomponentteihin. Tällä hetkellä organisaatiossa luodaan paljon komponentteja itse, mutta haastateltavien mukaan komponentteja voitaisiin myös ostaa valmiina tai toteutus voitaisiin ulkoistaa. Tämä komponentin luomisen tai hankinnan suunnittelu sisällytetään luotavaan toimintamalliin.

Haastatteluissa nousi myös esille tarve selkeälle dokumentoinnille. Dokumentoinnilta toivottiin selkeyttä erityisesti komponentin variaatiopisteisiin ja yleisesti komponentin toiminnallisuuteen. Organisaatiossa on jo tällä hetkellä pyritty dokumentoimaan erityisesti komponenttien variaatiopisteitä, mutta varsinaista selkeää ja sovittua prosessia ei tälle ole. Yksittäisten ohjelmistokomponenttien dokumentointi on monesti ohjelmistokehittäjän omalla vastuulla. Luotavaan toimintamalliin lisätään oma prosessi komponenttien dokumentoinnille, jotta kaikki komponentit tulisivat kattavasti dokumentoiduiksi.

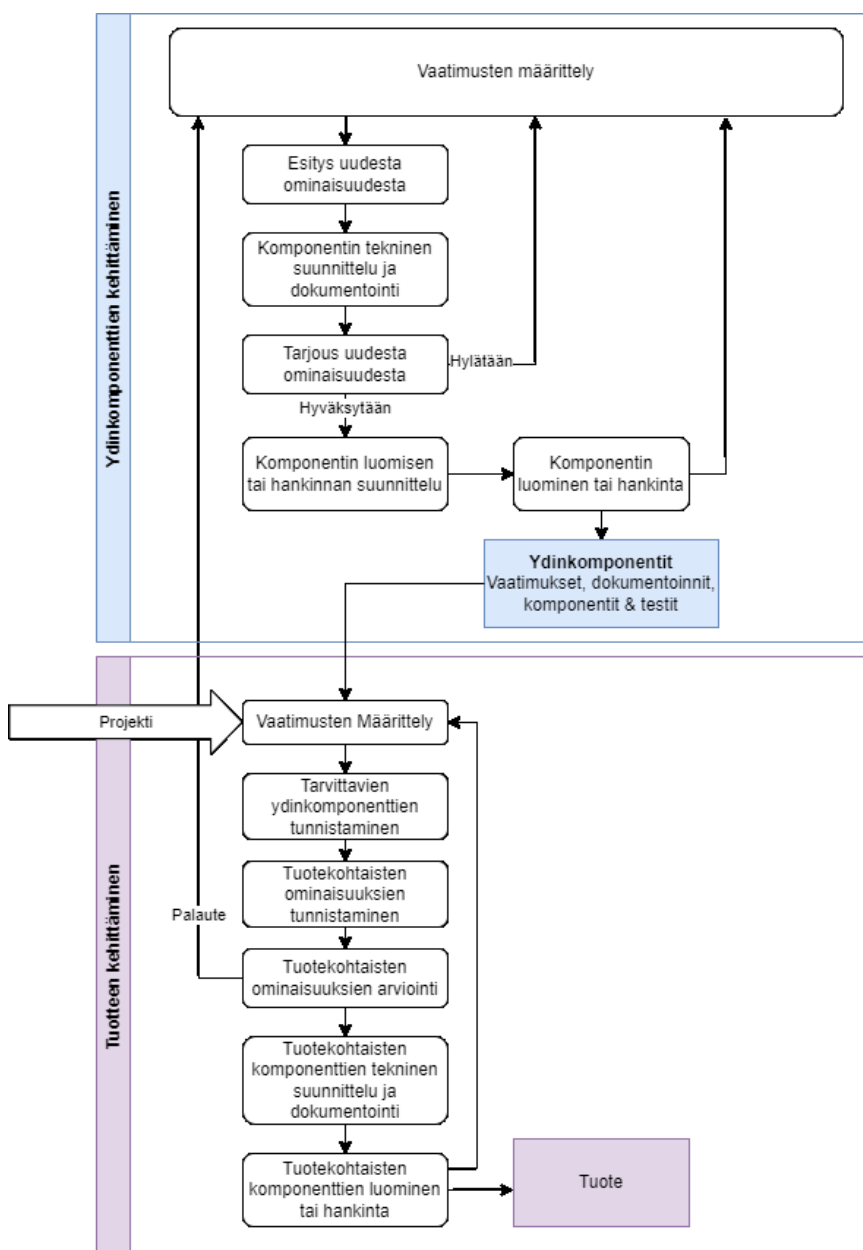
Tuotteen kehittämisen osalta organisaatiossa ei ollut niin paljoa valmiita toimintatapoja. Tällä hetkellä tuotteen kehittäminen alkaa organisaatiossa vaatimusten määrittelyllä, joka toteutetaan tuotteen toiminnallisten vaatimuksien pohjalta. Vaatimusten määrittelyssä pyritään tunnistamaan ne ominaisuudet, jotka voidaan toteuttaa ohjelmistotuotelinjan ydinkomponenteilla ja ominaisuudet, jotka vaativat uusia tuotekohtaisia komponentteja. Myös mahdolliset variaatiot ydinkomponenteista pyritään tunnistamaan tässä vaiheessa.

Uusien komponenttien osalta vaatimusten määrittelyssä tulee suorittaa myös määrittely komponentin teknisestä toteutuksesta. Haastattelujen perusteella tässä vaiheessa suoritetaan myös arvio siitä, että onko komponentille tar-

vetta myös tulevaisuudessa ja tulisiko se lisätä myös ydinkomponentteihin. Tärkeä osa toimintamallia on siis palaute tuotteen kehittämisen ydinkomponenttien kehittämiseen.

Uuden tuotekohtaisen komponentin osalta vaatimusten määrittelyn jälkeen tehdään päätös komponentin toteuttamisesta. Vaihtoehtoina on, että komponentti luodaan organisaation sisällä, luominen voidaan ulkoistaa tai komponentti voidaan ostaa valmiina. Arvio tehdään resurssien ja komponentin käyttökohteen perusteella.

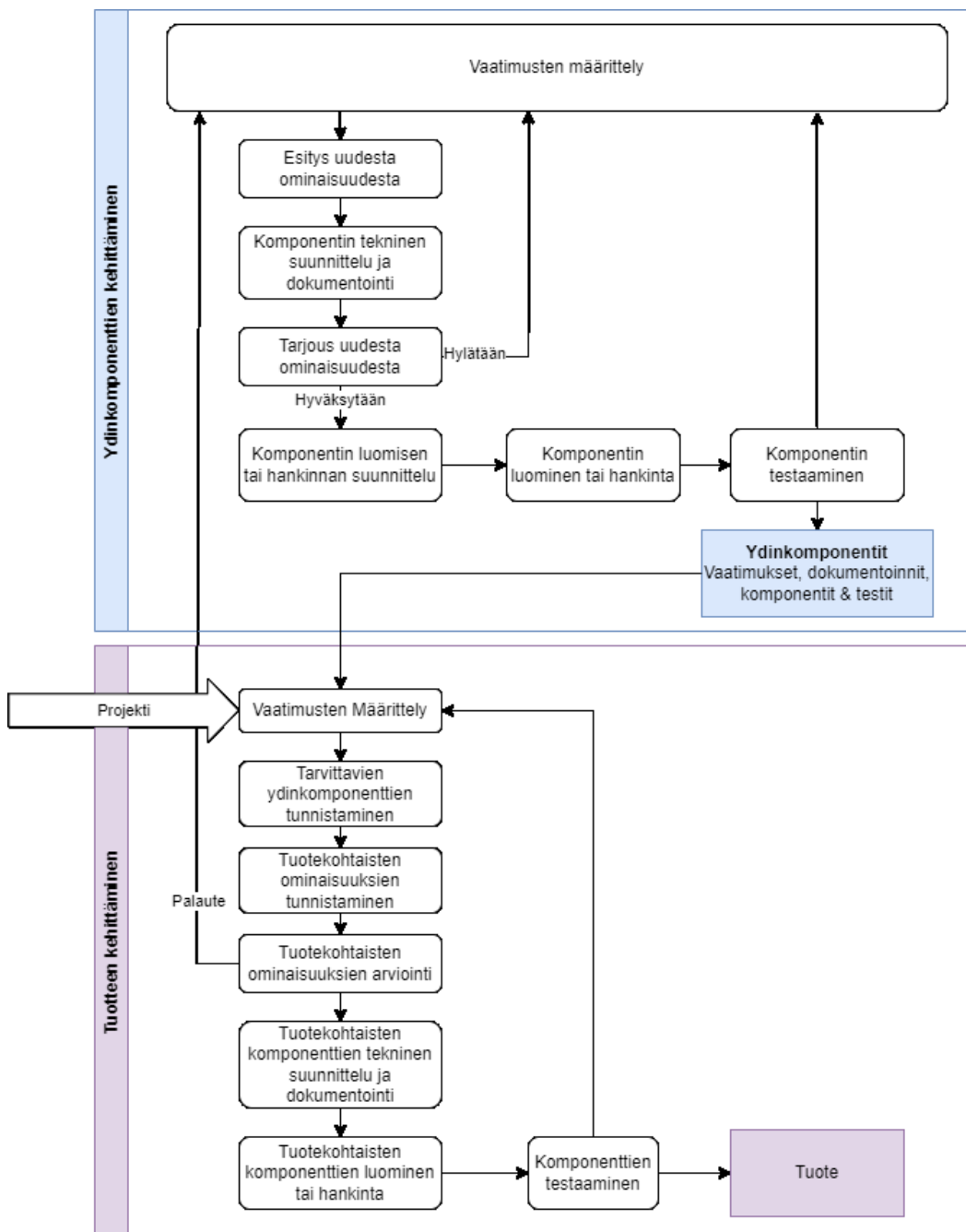
Organisaatiossa on haastattelujen perusteella selkeitä mielipiteitä ja joitakin käytäntöjä ohjelmistotuotelinjan tuotteen kehittämiseen. Näitä saadaan sisällytettyä luotavaan toimintamalliin, jotta organisaation totut tavat näkyvät toimintamallissa. Toimintamalli toisen iteraation jälkeen esiteltynä kuviossa 8.



KUVIO 8 Toimintamalli toisen iteraation jälkeen.

### 6.2.3 Kolmas iteraatio kirjallisuuden täydennyksistä

Toimintamallin luomisen kolmannessa iteraatiossa toimintamalliin lisätään täydennyksiä kirjallisuudesta. Toimintamalliin täydennetään kirjallisuudesta tunnistettuja tärkeitä ja olennaisia prosesseja ohjelmistotuotelinjan toiminnasta. Haastatteluista tunnistettiin jo monia tärkeitä kirjallisuudessa esiintyneitä prosesseja, mutta esimerkiksi haastattelujen rajallisen ajan vuoksi joitain prosesseja on voinut unohtua mainita. On myös mahdollista, että organisaatiossa ei ole huomioitu kaikkia kirjallisuudessa esiintyviä seikkoja. Täydennetty toimintamalli esitettyä kuviossa 9.



KUVIO 9 Toimintamalli kolmannen iteraation jälkeen.

Toimintamalli sisälsi jo toisen iteraation jälkeen monia kirjallisuudessa esitettyjä prosesseja ohjelmistotuotelinjan toimintaan. Yhtenä keskeisenä puutteena kirjallisuuteen verrattuna on kuitenkin testaaminen. Haastatteluissa ei noussut esille testaamisen rooli ohjelmistotuotelinjan toiminnassa. Yksi haastateltavista mainitsi kuitenkin testaamisen olevan tärkeää. Tarkemmin testaamista ei kuitenkaan käsitelty missään haastatteluissa.

Kirjallisuudessa testaaminen on nostettu merkittävään rooliin. Esimerkiksi Van der Linden, Schmid ja Rommes (2007) ja Clements ja Northrop (2002) mainitsevat viitekehyksissään testaamisen tärkeyden niin ydinkomponenttien kuin tuotteenkin kehittämisessä. Ydinkomponenttien kehittämisessä komponenttien testaaminen on tärkeää, jotta niiden laatu ja toimivuus voidaan varmistaa. Haasteena tässä on kuitenkin ydinkomponenttien sisältämät variaatiopisteet. Tuotetta kehittäessä testaamista tulisi keskittää erityisesti uusiin tuotekohtaisiin ominaisuuksiin ja kaikkien komponenttien yhteensopivuuden testaamiseen. Toimintamalliin lisättiin siis kirjallisuuden pohjalta prosessi testaamiselle. Prosessi lisättiin ydinkomponenttien ja tuotteen kehittämisen prosesseihin.

### 6.3 Toimintamalli

Luotu toimintamalli jakaa ohjelmistotuotelinjan toiminnan kahteen eri pääprosessiin. Nämä pääprosessit ovat ydinkomponenttien kehittäminen ja tuotteen kehittäminen. Ydinkomponenttien kehittäminen keskittyy kehittämään ohjelmistotuotelinjan ydinkomponentteja eli sen alustaa. Tuotteen kehittäminen puolestaan keskittyy tuotteiden valmistamiseen ohjelmistotuotelinjan ydinkomponenttien ja tuotekohtaisten komponenttien pohjalta. Luotu toimintamalli on esitettyinä kuviossa 9.

Toimintamallissa ydinkomponenttien kehittämisen ensimmäinen ja keskeisin prosessi on vaatimusten määrittely. Vaatimusten määrittely on jatkuva prosessi, jossa pyritään pohtimaan vaatimuksia ohjelmistotuotelinjan ydinkomponenteille. Vaatimusten määrittelyyn osallistuvat ohjelmistotuotelinjan toiminnan kannalta keskeisimmät sidosryhmät eli ydinkomponentteja kehittävä kehitystiimi, tuotteita kehittävä projektitiimi ja organisaation tuotteita myyvä myyntitiimi. Näin vaatimusten määrittelyyn saadaan mielipiteitä kaikilta keskeisiltä sidosryhmiltä ja ohjelmistotuotelinjan ydinkomponenttien vaatimukset vastaavat eri sidosryhmien tarpeita.

Seuraavassa ydinkomponenttien kehittämisen vaiheessa ohjelmistotuotelinjan tuoteomistaja tekee esityksen ydinkomponentteihin lisättävästä ominaisuudesta. Tuoteomistaja tekee ehdotuksensa vaatimusten määrittelyn pohjalta. Esitys ominaisuudesta tehdään ydinkomponentteja kehittäväälle kehitystiimille. Esityksessä käydään läpi lisättävän ominaisuuden keskeisimmät tiedot ja toiminnalliset vaatimukset.

Esityksen jälkeen kehitystiimi pitää oman sisäisen palaverin lisättävästä ominaisuudesta. Palaverissa kehitystiimi suunnittelee tarkemmin lisättävän ominaisuuden toiminnallisuutta ja teknistä toteutusta. Tässä vaiheessa ominaisuudesta tehdään myös tarkka dokumentointi sen toiminnallisuudesta ja mahdollisista variaatiopisteistä. Sisäisen palaverin jälkeen kehitystiimi tekee oman ehdotuksensa lisättävästä ominaisuudesta. Tuoteomistaja tekee ehdotuksen pohjalta päätöksen siitä, että lisätäänkö ominaisuus ydinkomponentteihin.

Mikäli kehitystiimin ehdotus hylätään, prosessia ei jatketa kyseisen ominaisuuden osalta eteenpäin, vaan prosessi palaa takaisin vaatimusten määrittelyyn. Jos ehdotus hyväksytään, aletaan ominaisuuden osalta suunnittelemaan sen toteuttamista tai hankintaa. Ominaisuus voidaan toteuttaa organisaation sisällä, toteuttaminen voidaan ulkoistaa tai ominaisuus voidaan ostaa valmiina komponenttina. Päätös ominaisuuden luomis- tai hankintatavasta tehdään ominaisuuden käyttökohteen ja sen vaatimien resurssien perusteella. Organisaation ydinosaamiseen liittyvät ominaisuudet pyritään toteuttamaan organisaation sisällä ja muita ominaisuuksia voidaan mahdollisesti ulkoistaa tai ostaa.

Seuraavana toimintamallin vaiheena on lisättävän ydinkomponentin luominen tai hankinta edellisessä vaiheessa päätetyllä tavalla. Organisaation kannalta työläin komponentin hankintatapa on luoda se itse. Luomalla komponentti itse voidaan kuitenkin varmistua siitä, että esimerkiksi organisaation keskeistä ydinosaamista voidaan hyödyntää komponentin luomisessa. Tällöin komponentista saadaan juuri sellainen kuin halutaan.

Uusien komponenttien kehittämisessä keskeisessä roolissa on myös toimintamallin seuraava vaihe eli testaaminen. Toimintamallissa komponentin luominen ja testaaminen on esitetty peräkkäisessä järjestyksessä, mutta todellisudessa kehitystyötä ja testaamista toteutetaan hyvin samanaikaisesti. Komponenttia ei siis luoda kokonaan valmiiksi ja testausta suoritetaan vain sen jälkeen valmiille komponentille vaan testausta pyritään suorittamaan jo kehitystyön aikana, jotta varmistetaan komponentin täyttävän sille asetetut vaatimukset. Kehitystyön ja testaamisen jälkeen ohjelmistotuotelinjan ydinkomponentti on valmis ja ydinkomponenttien kehittämisprosessi siirtyy takaisin alkuun eli vaatimusten määrittelyyn.

Tuotteen kehittämisen prosessi saa alkunsa uudesta projektista, jossa toteutetaan tuote asiakkaalle. Toimintamallissa tuotteen kehittämisen prosessi alkaa vaatimusten määrittelyllä. Tuotteen vaatimukset määritellään myyntiosastolta saatujen toiminnallisten vaatimuksien pohjalta. Vaatimusten määrittelyn jälkeen pyritään vaatimusten pohjalta tunnistamaan tuotteeseen tarvittavat ydinkomponentit ja tuotetta varten tarvittavat uudet tuotekohtaiset komponentit. Kun uudet tuotekohtaiset komponentit on tunnistettu, suoritetaan niille arviointi sen osalta, että onko niille käyttöä myös tulevaisuudessa. Mikäli komponentille voisi olla tarvetta myös tulevissa tuotteissa, voidaan komponentti toteuttaa ohjelmistotuotelinjan ydinkomponentteihin.

Seuraavana vaiheena on tuotekohtaisen komponentin tarkempi tekninen suunnittelu. Tässä vaiheessa käydään läpi komponentin vaatimukset ja muodostetaan sen pohjalta tarkka tekninen suunnitelma sen toteutuksesta. On myös

mahdollista, että komponentin toteuttaminen päätetään ulkoistaa tai komponentti voidaan ostaa valmiina. Tässä vaiheessa myös komponentin toteutus ja toiminnallisuus dokumentoidaan tarkasti.

Komponentin tarkemman suunnittelun ja dokumentoinnin jälkeen vuorossa on komponentin luominen tai hankinta. Kuten ydinkomponenttien kehittämisessäkin, myös tuotteen kehittämisessä komponenttien luominen ja testaaminen ovat hyvin samanaikaiset prosessit. Testaamista toteutetaan siis jo komponentin luomisen aikana.

Kun komponentin luominen ja testaaminen on saatu valmiiksi, on yksi tuotetta varten luotu komponentti valmis. Lopulta tuotekohtaisista uusista komponenteista sekä ohjelmistotuotelinjan ydinkomponenteista saadaan valmistettua asiakkaan vaatimuksia vastaava tuote.



## 7 TOIMINTAMALLIN ARVIOINTI

Tässä luvussa arvioidaan luotu toimintamalli. Arviointia varten pyydettiin palautetta ja mielipiteitä neljältä henkilöltä kohdeorganisaatiossa. Aluksi käydään läpi henkilöiden yleiset kommentit ja mielipiteet ja tämän jälkeen käydään läpi mahdolliset kehitysideoita, toimintamallista puuttuvat asiat ja haastateltavien muut kommentit toimintamallista. Tämän luvun tarkoituksena on siis validoida, että luotu toimintamalli soveltuisi tutkittavan organisaation käyttöön.

### 7.1 Yleiset arviot ja mielipiteet

Toimintamallin arviointiin osallistui neljä henkilöä organisaatiosta. Arviointia suorittavat henkilöt ovat organisaation tuotekehitys- ja ohjelmistokehitystii-meistä. Henkilöt ovat siis keskeisissä rooleissa ohjelmistotuotelinjan ydinkomponenttien sekä tuotteen kehittämisen prosesseissa. Haastateltavat henkilöt ovat henkilöitä, jotka voisivat mahdollisesti käyttää luotua toimintamallia.

Toimintamallin arvioinnissa haastateltavilta kysyttiin aluksi heidän mielipidettään siitä, että soveltuuko toimintamalli organisaation ohjelmistotuotelinjalle. Kaikki haastateltavat ilmaisivat, että heidän mielestään toimintamalli olisi soveltuva organisaation ohjelmistotuotelinjalle. Eräs haastateltava toteaaakin, että toimintamalli muistuttaa organisaatiossa jo aikaisemmin kokeiltua toimintamallia organisaation aikaisemman WCS-classic ohjelmiston osalta.

”Projektipuolen toiminta on ennenkin ollut tämän tyylinen. Me olemme siis joskus tehneet näin WCS-classicin kanssa. Yritetty tunnistaa ominaisuuksia, jotka voivat olla ydinkomponentteja. Olen sitä mieltä, että niin se kuuluisi tehdäkin, eli tähän malliin.” (Henkilö 4, haastattelu, 9.10.2023)

Toimintamallin soveltuvuuden osalta yksi haastateltavista kuitenkin nosti esille toimintamallin yhteensopivuuden organisaation muihin toimintatapoihin.

Hänen mielestään toimintamalli siis on soveltuva ohjelmistotuotelinjan toimintaan, mutta voidaanko toimintamallia käyttää yhdessä esimerkiksi organisaatiossa hyödynnettävien ketterän kehityksen toimintamallien kanssa.

”Cimcorpissa on käytössä tietynlainen Scrum-toimintamalli ohjelmistokehitykseen, joka on tietynlaiseen prosessiin mietitty. Miten luotu toimintamalli istuu tämän kanssa yhteen.” (Henkilö 3, haastattelu, 1.11.2023)

Toisena kysymyksenä haastateltavilta kysyttiin, että olisiko toimintamallilla positiivinen vaikutus ohjelmistotuotelinjan toimintaan. Kaikki haastateltavat olivat sitä mieltä, että toimintamallilla olisi positiivinen vaikutus ohjelmistotuotelinjan toimintaan. Haastateltavat kommentoivat, että aikaisemmassa toiminnassa ei ole ollut selkeää toimintamallia, jolloin esimerkiksi projekteissa luodut ominaisuudet eivät ole ikinä päätyneet osaksi ydinkomponentteja.

”Ydinkomponentteja ei ole WCS-classic puolelle muodostunut viimeiseen viiteen vuoteen, eli se ei ainakaan toimi. Eli olen sitä mieltä, että tämä malli toimisi. Ongelmana ollut, että projektit ajelehtivat yksinään, mikään ei kerää ydinkomponentteja alustaan.” (Henkilö 4, haastattelu, 9.10.2023)

”Tällä hetkellä ei ole käytössä mitään mallia ja jälki on sen näköistä. Prosessi ei toimi.” (Henkilö 4, haastattelu, 9.10.2023)

Haastateltavilta kysyttiin myös, että käyttäisivätkö he itse toimintamallia. Jokainen haastateltava ilmoitti, että he voisivat käyttää toimintamallia. Yksi haastateltavista kommentoi, että toimintamallia voitaisiin myös hyödyntää organisaation aikaisemman ohjelmiston WCS-classicin toiminnassa.

”Hyvä ja selkeä esitysmalli, jota voitaisiin myös hyödyntää Cimcorp classic WCS toteutuksissa.” (Henkilö 6, haastattelu, 24.10.2023)

”Voisin lähteä käyttämään, mikäli yrityksemme haluaa sitoutua tällaisen mallin käyttämiseen.” (Henkilö 3, haastattelu, 1.11.2023)

Yleisellä tasolla siis arviointien pohjalta voidaan todeta, että tutkimuksessa luotu toimintamalli soveltuisi hyvin tutkittavan organisaation ohjelmistotuotelinjan toimintaan. Toimintamalli voisi arviointien perusteella myös vaikuttaa positiivisesti ohjelmistotuotelinjan toimintaan.

## 7.2 Kehitysideat, puuttuvat asiat ja muut kommentit

Kokonaisuudessaan haastateltavat olivat sitä mieltä, että toimintamalli on sellaisenaan soveltuva organisaation ohjelmistotuotelinjan toimintaan. Haastateltavilta kuitenkin saatiin joitain kehitysideoita ja kommentteja toimintamallista. Varsinaisia puutteita haastatteluissa ilmeni vain yksi.

Yksi haastateltavista kommentoi puutetta ydinkomponenttien kehittämisessä. Haastateltava ehdotti, että toimintamallissa olisi polku myös uuden ominaisuuden poistamiseksi vaatimusten määrittelystä. Eli kun tarjous uudesta ominaisuudesta on hylätty, niin olisi myös mahdollista poistaa se kokonaan, esimerkiksi jos se on todettu tarpeettomaksi. Toimintamallissa ydinkomponentteihin tarjottu hylätty ominaisuus palaa aina vaatimusten määrittelyyn.

Haastateltavilta saatiin useita ideoita ja mahdollisia parannuksia toimintamalliin. Yksi mahdollinen parannus olisi se, että tuotteen kehittämisen prosessissa myös ydinkomponenttien kehittäjät osallistuisivat tuotteeseen vaadittavien ydinkomponenttien tunnistamiseen. Haastateltava toi esille sen, että tarvittavien ydinkomponenttien tunnistaminen voi olla hyvin haastavaa ja se vaatii paljon tietoa ja osaamista ohjelmistotuotelinjan ydinkomponenteista. Tuotteen kehittämisen tiimillä yksinään ei välttämättä ole tarpeeksi osaamista tunnistaa kaikkia ydinkomponentteja.

”Ideaalitalanteessa meillä on projektissa aina senioritason softakehittäjiä. Ydinkomponenttien tunnistaminen vaatii tietotaitoa ja tuntemusta meidän projekteista.” (Henkilö 4, haastattelu, 9.10.2023)

Ydinkomponenttien kehittäjien ja tuotteen kehittäjien välinen yhteistyö nousi haastatteluissa esille myös tuotekohtaisten komponenttien arvioinnissa. Jos tuotetta varten joudutaan kehittämään hyvin monimutkainen tai työläs komponentti, mikä voisi olla myös ohjelmistotuotelinjan ydinkomponentti, niin miten työnjako hoidetaan komponentin toteutuksessa. Haastateltava pohti, miten tuotteen kehittämisen projektitiimillä riittää resurssit tuollaisen komponentin toteuttamiseen.

”Ei se ole niin mustavalkoinen, että jos siellä on uusi ominaisuus alustaan, niin tuotekehitystiimi tekee sen tai, jos se on projektikohtainen, niin projektitiimi tekee sen. Tarvitsee ne yhdessä käydä läpi. Tuki tarvitaan tuotekehityksen tiimiltä.” (Henkilö 4, haastattelu, 9.10.2023)

Toinen esille noussut parannus toimintamalliin olisi se, että toimintamallissa kävisi selkeämmin ilmi se, miten jo toteutetuista tuotteista löytyviä komponentteja tuodaan ydinkomponenteiksi. Tämä ei toimintamallissa näy selkeästi, mutta vanhoja komponentteja pyritään hyödyntämään ydinkomponenttien luomisessa ja hankinnassa.

Yksi arvioinneissa esiin noussut parannusehdotus liittyy tuotteen kehittämisessä kehitettävien komponenttien tekniseen suunnitteluun ja dokumentointiin. Yksi haastateltavista totesi, että tuossa vaiheessa prosessia olisi myös hyvä olla palautekanava ydinkomponenttien vaatimusten määrittelyyn, sillä tuossa vaiheessa komponenttia on ehditty suunnitella tarkemmin ja mahdollinen tarve uudelle ydinkomponentille on voinut syntyä.

Palautekanava tuotteen kehittämisen muista vaiheista ydinkomponenttien kehittämiseen nousi esille myös toisessa haastattelussa. Toisen haastateltavan

mielestä palautekanava voisi olla auki kaikissa tuotekohtaisen ominaisuuden kehittämiseen ja testaamiseen liittyvissä prosesseissa. Tämä olisi hänen mielestään hyvä, koska ominaisuus voi vielä muuttua merkittävästikin eri vaiheiden aikana.

”Matka tuotekohtaisen ominaisuuden arvioinnista toteutukseen, toteutuksesta testaamiseen ja lopulta edes julkaisukelpoiseen ominaisuuteen on pitkä.” (Henkilö 4, haastattelu, 9.10.2023)

Toimintamallia kommentoitiin myös sen ”vesiputous”-tyylin osalta. Yksi haastateltavista huomautti, että ohjelmistokehityksessä prosessit eivät aina etene vesiputousmallin mukaisesti järjestyksessä, vaan eri prosessien välillä voidaan liikkua muutenkin. Esimerkiksi komponentin testaamista voidaan suorittaa jo sen kehitystyön aikana. Haastateltava kuitenkin toteaa, että toimintamallissa on ohjelmistokehityksen kannalta oikeat prosessit ja malli voi toimia myös tuollaisenaan.

”Hieman vesiputousmallin näköinen kokonaisuudessaan. Eihän ne softa-prosessit aina mene näin suoraan... Mutta kyllä se noinkin voi toimia.” (Henkilö 4, haastattelu, 9.10.2023)

Yksi haastateltavista nosti esille ydinkomponenttien vaatimusten määrittelyn tärkeyden. Toimintamallissa vaatimusten määrittelyä ei ole avattu juurikaan. Toimintamallista ei siis käy ilmi mitä kaikkea vaatimusten määrittely oikeasti sisältää. Haastateltavan mielestä vaatimusten määrittelyä olisi voinut avata enemmän, mutta samalla toimintamallista tulisi huomattavasti sekavampi. Mielipide pohjautuu hyvin vahvasti organisaation ongelmaan vaatimusten määrittelyssä. Kuten haastattelujen analysoinnin luvussa todettiin, vaatimusten määrittely on todettu olevan ongelmakohta ohjelmistotuotelinjan toiminnassa.

”Koko tämän prosessin onnistumiselle edellytyksenä on onnistunut vaatimusten määrittely ja niiden kuvaaminen.” (Henkilö 3, haastattelu, 1.11.2023)

Yleisesti ottaen kaikki haastateltavat pitivät toimintamallia toimivana ja voisivat käyttää toimintamallia. Haastateltavien mielestä toimintamallilla voisi myös olla positiivisia vaikutuksia ohjelmistotuotelinjan toimintaan. Haastateltavat eivät maininneet toimintamallista merkittäviä puutteita. Selkein puute toimintamallissa haastateltavien mielestä oli kattavampi palautteen antaminen tuotteen kehittämisestä ydinkomponenttien kehittämiseen.

Haastateltavilta tuli kuitenkin muitakin hyviä lisäideoita ja kommentteja toimintamallista. Monet lisäideat edellyttäisivät toimintamallien prosessien avaamista tarkemmalle tasolle. Näiden pohjalta toimintamallia voisi kehittää eteenpäin, mikäli toimintamallin prosesseja haluaa tarkentaa syvemmälle tasolle. Esimerkiksi ohjelmistotuotelinjan vaatimusten määrittely on yksissään jo hyvin monimutkainen prosessi, jonka tarkempia vaiheita voisi tutkia. Haastateltavien mielestä kuitenkin laajemmalla tasolla toimintamalli on parempi pitää selkeänä ja yksinkertaisena. Liian tarkka toimintamalli voi olla vaikeaa ja työlästä tulkita. Arvioinnit tiivistettynä taulukossa 3.

Taulukko 3 Yhteenvedo toimintamallin arvioinnista

Henkilö	3	4	5	6
Soveltuvuus?	Kyllä	Kyllä	Kyllä	Kyllä
Positiivinen vaikutus ohjelmistotuotelinjan toimintaan?	Kyllä	Kyllä	Kyllä	Kyllä
Lisäideoita?	-	-	-	Mallin yhteensopivuus organisaatiossa käytössä olevaan Scrum-malliin.
Puuttavia asioita?	-	-	Tarjottavan ominaisuuden hylkääminen kokonaan tarjouksen esittämisen jälkeen.	Palaute kanava ydinkomponenttien kehittämiseen myös tuotteen kehittämisen myöhemmistä vaiheista.
Parannuksia?	Ydinkomponenttien kehittäjien avustus tuotteen kehittämiseen tarvittavien ydinkomponenttien tunnistamisessa ja uusien komponenttien luomisessa.	Jo toteutettujen tuotteiden sisältämien komponenttien päätyminen ydinkomponentteihin.	Palaute kanava tuotekohtaisen komponentin teknisestä suunnittelusta ja dokumentoinnista ydinkomponenttien vaatimusten määrittelyyn.	Vaatimusten määrittelyn tarkempi kuvaus. Vaatimusten määrittely todella suuressa roolissa toimintamallin toimivuuden osalta.
Käyttäisikö toimintamallia?	Kyllä	Kyllä	Kyllä	Kyllä
Muita kommentteja	Vesiputousmalli ei aina realistinen ohjelmistokehityksessä. Myyntiosaston kanssa tehtävä yhteistyö tärkeää.	Tuotteen ja ydinkomponenttien kehittäjien tiimien yhteistyö tärkeää tuotekohtaisten komponenttien arvioinnissa.	Voitaisiin hyödyntää myös organisaation muiden ohjelmistojen toteutuksissa.	Malli on hyvä, mutta miten toimii käytännössä organisaation muiden toimintatapojen kanssa yhteen. Mikään malli ei ole valmis sellaisenaan, vaativat usein jatkokehitystä ja oppimista todellisuudesta.

## 8 JOHTOPÄÄTÖKSET

Tässä tietojärjestelmätieteen pro gradu-tutkielmassa tutkittiin tuotepohjaisen ohjelmistokehityksen hallintaa ja kehittämistä organisaatiossa. Tutkimuksessa tutkittiin Cimcorp Oy:n ohjelmistotuotelinjaa. Ohjelmistotuotelinja on kustannustehokas keino tuottaa yksilöityjä ja laadukkaita ohjelmistotuotteita asiakkaille tietyn markkinasegmentin sisällä. Selkeällä toimintamallilla on merkittävä vaikutus ohjelmistotuotelinjan tuotteiden laatuun.

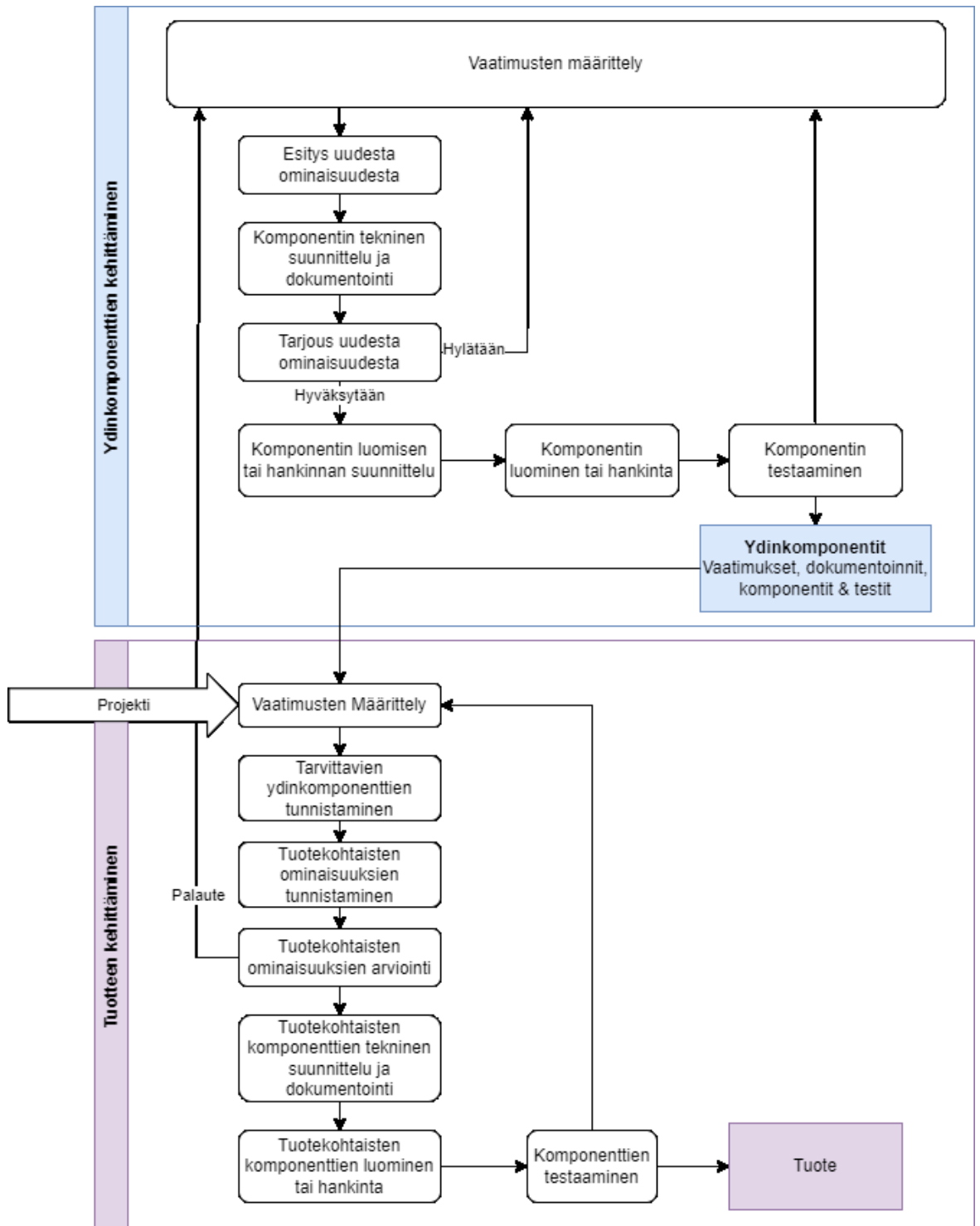
Tutkimus toteutettiin suunnittelutieteellisenä tutkimuksena. Tutkimuksessa hyödynnettiin organisaation ympäristöä ja aiemman tutkimuksen tietopohjaa toimintamallin luomiseen. Tietopohjaa varten toteutettiin kirjallisuuskatsaus, jolla perehdyttiin ohjelmistotuotelinjoihin ja niihin liittyviin prosesseihin. Organisaation ympäristöön tutustuttiin puolistrukturoiduilla haastatteluilla, joilla selvitettiin organisaation nykyisiä toimintatapoja ja käytäntöjä ohjelmistotuotelinjaan liittyen.

Kirjallisuuskatsauksessa huomattiin, että monet aikaisemmat tutkimukset hyödyntävät teoriapohjana kahta kattavaa teosta, jotka käsittelevät ohjelmistotuotelinjaan liittyviä prosesseja. Tässäkin tutkimuksessa ohjelmistotuotelinjan teoriaa tarkasteltiin pitkälti samojen teosten pohjalta. Toimintamallin luomisessa hyödynnettiin kirjallisuuden lisäksi organisaation nykyisiä toimintatapoja ja käytäntöjä. Toimintamallia luodessa kuitenkin huomattiin, että haastatteluissa esiin nousseet toimintatavat ja mielipiteet mukailivat pitkälti teoriapohjassa esiintyneitä prosesseja. Tutkimuksen tuloksena syntynyt toimintamalli esitettyinä kuviossa 10. Tutkimuksen tavoitteena oli muodostaa tutkittavalle organisaatiolle toimintamalli ohjelmistotuotelinjalle ja vastata tutkimuskysymykseen:

- Millaisella toimintamallilla tuotepohjaista ohjelmistokehitystä voidaan toteuttaa organisaatiossa?

Tutkimuksessa luodulle toimintamallille suoritettiin arviointi, jossa organisaation ohjelmistotuotelinjan sidosryhmiin kuuluvilta henkilöiltä pyydettiin mielipiteitä ja kommentteja toimintamallista. Toimintamallin arvioinnin mukaan

luotu toimintamalli olisi soveltuva organisaation ohjelmistotuotelinjalle. Kaikkien arvioiden mukaan ohjelmistotuotelinjalla voisi myös olla positiivisia vaikutuksia ohjelmistotuotelinjan toimintaan. Kaikki haastateltavat myös ilmoittivat, että he voisivat käyttää toimintamallia. Arvioinnin pohjalta voidaan siis vastata tutkimuskysymykseen, että tutkimuksessa luodulla toimintamallilla voitaisiin toteuttaa tuotepohjaista ohjelmistokehitystä organisaatiossa.



KUVIO 10 Organisaation ohjelmistotuotelinjalle luotu toimintamalli.

Tutkimuksen selkeimpänä rajoitteena on se, että tuotepohjaista ohjelmistokehitystä tutkittiin vain yhdessä organisaatiossa. Lisäämällä tutkittavien organisaatioiden määrää, haastatteluilla olisi voitu saada monimuotoisempia toimintatapoja ja käytäntöjä tuotepohjaisesta ohjelmistokehityksestä.

Toisena rajoitteena oli haastateltavien henkilöiden määrä. Tutkittava ohjelmistotuotelinja oli organisaatiossa suhteellisen uusi, eikä sen toimintaan ole osallistunut vielä kovinkaan paljon henkilöitä. Tämän takia haastatteluihin valittiin vain kolme henkilöä, jotka ovat olleet tekemisissä ohjelmistotuotelinjan kanssa. Suuremmalla haastateltavien määrällä oltaisiin voitu saada enemmän toimintatapoja ja mielipiteitä ohjelmistotuotelinjan toiminnasta.

Toimintamallin arvioinnissa rajoitteena oli se, että toimintamallin toimivuutta ei testattu käytännössä. Toimintamallia ei siis käytetty organisaation tuotepohjaisessa ohjelmistokehityksessä. Arviointi suoritettiin vain haastatteluilla ja mallin toimivuuden arviointi pohjautuu siis vain haastateltujen henkilöiden mielipiteisiin. Toimintamallin testaaminen käytännössä ei olisi ollut mahdollista aikataulun tai resurssien puolesta.

Mahdollisia jatkotutkimusaiheita aiheelle olisi esimerkiksi toimintamallin eri prosessien tarkempi kuvaaminen. Esimerkiksi Ohjelmistotuotelinjan ydinkomponenttien vaatimusten määrittelyä voisi tutkia tarkemmin. Haastatteluissa nousi esille, että vaatimusten määrittely on todella suuri haaste organisaatiossa. Vaatimusten määrittelyyn liittyvää prosessia voisi siis olla mielekästä tutkia tarkemmin.

Toinen mahdollinen tutkimusaihe nousi esille toimintamallin arvioinnissa. Yksi haastateltavista pohti toimintamallin yhteensopivuutta toisten toimintamallien kanssa. Toimintamallin toimivuutta siis esimerkiksi, vaikka Scrum-mallin kanssa voisi tutkia. Tämän tutkiminen voisi tuoda lisäarvoa toimintamallille ja mahdollistaisi toimintamallin konkreettisen hyödyntämisen muiden toimintamallien kanssa.



## LÄHTEET

- Adams, W. C. (2015). Conducting semi - structured interviews. *Handbook of practical program evaluation*, 492-505.
- Apel, S., Batory, D., Kästner, C., & Saake, G. (2016). Feature-oriented software product lines (pp. 19-26). Springer-Verlag Berlin An.
- Bagheri, E., Ensan, F., & Gasevic, D. (2012). Decision support for the software product line domain engineering lifecycle. *Automated Software Engineering*, 19, 335-377.
- Bockle, G., Clements, P., McGregor, J. D., Muthig, D., & Schmid, K. (2004). Calculating ROI for software product lines. *IEEE software*, 21(3), 23-31.
- Bosch, J. (2001, May). Software product lines: organizational alternatives. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001* (pp. 91-100). IEEE.
- Clements, P. (1999). Software Product Lines: A New Paradigm for the New Century. *Crosstalk*, 12(2), 20-22.
- Clements, P., & Northrop, L. (2002). *Software product lines*. Boston: Addison-Wesley.
- do Carmo Machado, I., McGregor, J. D., & Santana de Almeida, E. (2012). Strategies for testing products in software product lines. *ACM SIGSOFT Software Engineering Notes*, 37(6), 1-8.
- Engström, E., & Runeson, P. (2011). Software product line testing—a systematic mapping study. *Information and Software Technology*, 53(1), 2-13.
- Grüner, S., Burger, A., Kantonen, T., & Rückert, J. (2020, October). Incremental migration to software product line engineering. In *Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A-Volume A* (pp. 1-11).
- Hallsteinsen, S., Hinchey, M., Park, S., & Schmid, K. (2008). Dynamic software product lines. *Computer*, 41(4), 93-95.
- Harvey-Jordan, S., & Long, S. (2001). The process and the pitfalls of semi-structured interviews. *Community Practitioner*, 74(6), 219.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly*, 28(1), 6.
- Jones, L. G., & Soule, A. L. (2002). *Software process improvement and product line practice: CMMI and the framework for software product line practice*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

- Kang, K. C., Sugumaran, V., & Park, S. (Eds.). (2009). Applied software product line engineering. CRC press.
- Knauber, P., Bermejo, J., Böckle, G., Leite, J. C. S. D. P., van der Linden, F., Northrop, L., ... & Weiss, D. M. (2002). Quantifying product line benefits. In Software Product-Family Engineering: 4th International Workshop, PFE 2001 Bilbao, Spain, October 3-5, 2001 Revised Papers 4 (pp. 155-163). Springer Berlin Heidelberg.
- Krueger, C. (2002, April). Easing the transition to software mass customization. In Software Product-Family Engineering: 4th International Workshop, PFE 2001 Bilbao, Spain, October 3-5, 2001 Revised Papers (pp. 282-293). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Kvale, S., & Brinkmann, S. (2009). Interviews: Learning the craft of qualitative research interviewing. sage.
- Metzger, A., & Pohl, K. (2014). Software product line engineering and variability management: achievements and challenges. Future of software engineering proceedings, 70-84.
- Montagud, S., Abrahão, S., & Insfran, E. (2012). A systematic review of quality attributes and measures for software product lines. Software Quality Journal, 20, 425-486.
- Northrop, L. M. (2002). SEI's software product line tenets. IEEE software, 19(4), 32-40.
- Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., ... & Little, R. (2007). A framework for software product line practice, version 5.0. SEI.-2007-<http://www.sei.cmu.edu/productlines/index.html>.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of management information systems, 24(3), 45-77.
- Pohl, K., & Metzger, A. (2018). Software product lines. The Essence of Software Engineering, 185-201.
- Pohl, K., Böckle, G., & Van Der Linden, F. (2005). Software product line engineering (Vol. 10, pp. 3-540). Heidelberg: Springer.
- Reinhartz-Berger, I., Cohen, S., Bettin, J., Clark, T., & Sturm, A. (2013). Domain engineering. Heidelberg: Springer.
- Schaefer, I., & Hähnle, R. (2011). Formal methods in software product line engineering. Computer, 44(02), 82-85.
- Schmid, K. (2001, October). An assessment approach to analyzing benefits and risks of product lines. In 25th Annual International Computer Software and Applications Conference. COMPSAC 2001 (pp. 525-530). IEEE.
- Sugumaran, V., Park, S., & Kang, K. C. (2006). Software product line engineering. Communications of the ACM, 49(12), 28-32.

- Tüzün, E., Tekinerdogan, B., Kalender, M. E., & Bilgen, S. (2015). Empirical evaluation of a decision support model for adopting software product line engineering. *Information and Software Technology, 60*, 77-101.
- Van der Linden, F. J., Schmid, K., & Rommes, E. (2007). *Software product lines in action: the best industrial practice in product line engineering*. Springer Science & Business Media.