



Vaasan yliopisto
UNIVERSITY OF VAASA

Juhana Kiviniemi

Dataputken suunnittelu DataOps määritelmän avulla

Tekniikan ja innovaatiojohtamisen yksikkö
Tietojärjestelmätieteen pro gradu -tutkielma
Tietojärjestelmätiede

Vaasa 2022

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen yksikkö**

Tekijä:	Juhana Kiviniemi		
Tutkielman nimi:	Dataputken suunnittelu DataOps määritelmän avulla		
Tutkinto:	Kauppateiden maisteri		
Oppiaine:	Tietojärjestelmätiede		
Työn ohjaaja:	Tero Vartiainen		
Valmistumisvuosi:	2022	Sivumäärä:	73

TIIVISTELMÄ:

Dataan perustuva toimiva ja luotettava päätöksentekokyvykyys on noussut yhdeksi merkittävimmäksi lisäarvon tuottajaksi moderneille nykypäivän yrityksille. Jatkuvasti muuttuva kilpailuympäristö yhdistettynä alati vaihtuviin mieltymyksiin asiakkaiden valinnoissa, pakottavat organisaatioita tekemään tietoisia mutta mahdollisimman nopeita dataan perustuvia liiketoimintapäätöksiä. Lisäksi tiedon kasvava rooli kilpailutekijänä on näkynyt yritysten strategisella tasolla, mikä on johtanut investointipääomien siirtymiseen kohti data-analytiikan kyvykkyysien kasvattamista. Valitettavasti kokemukset näiden investointien arvon lisäyksistä ovat jääneet alhaisiksi useiden organisaatioiden kohdalla. Yritysten sisäiset kommunikointiongelmien, organisaation kulttuuriin juurtuneet käytänteet sekä esimerkiksi teknologisten prosessien toimimattomuus jatkuvasti kehittyvän ja nopeasti toimitettavan analytiikan tuottamiseksi ovat muodostuneet esteiksi lisäarvon saavuttamiseksi. Näiden ongelmien ratkaisemiseksi on hiljattain muodostunut ajatus nimeltä DataOps, jossa joukolla parhaita käytänteitä hyödyntämällä pyritään vastaamaan data-analytiikan kohtaamiin haasteisiin. Tämän tutkielman tarkoituksena on selvittää mitä DataOps määritelmällä tarkoitetaan, mitkä ovat sen tyypillisimmät teknologiaan liitettävät ominaisuudet, ja kuinka näihin ominaisuuksiin perustuen voidaan suunnitella datan siirtämiseen, muokkaamiseen sekä lataamiseen tarkoitettu prosessi eli dataputki.

Tutkimuksen teoreettinen viitekehys muodostettiin kirjallisuudesta löydettyjen DataOpsin kuvauksien pohjalta, joiden voidaan nähdä lainaavan piirteitä DevOpsin, ketterän ohjelmistokehityksen sekä lean-ajattelun teorioista. Viitekehykseen perustuen tutkielman lopputuloksena luotiin suunnittelutieteellisellä tutkimuksella artefakti eli ylätasoinen ominaisuuslista DataOpsin tyypillisimmistä piirteistä, kun tavoitteena on kehittää datan siirtämiseen, muokkaamiseen ja lataamiseen kohdistettu prosessi. Artefaktin käyttöä demonstroitiin rakentamalla siihen pohjautuva DataOpsin periaatteita mukaileva dataputki, jonka suoritusta simuloitiin kehitys- ja testausympäristöissä.

Määritelmänä DataOps on suhteellisen uusi, ja siitä löytyvä tieteellinen tutkimus vielä vähäistä. Käsitettä voidaan kuvailla eri tavoilla, painottaen joko teknologisia aiheita tai huomioimalla enemmän ihmisiin sekä organisatorisiin tekijöihin kohdistuvia alueita. Vaikka yhtä yksiselitteistä määritelmää ei vielä ole syntynyt, kaikki löydetty kuvaukset muistuttivat toisiaan eikä niiden välillä havaittu merkittäviä ristiriitoja. DataOps voidaan nähdä joukkona parhaita käytänteitä, joka lainaa piirteitä DevOpsin, ketterän ohjelmistokehityksen sekä lean-ajattelun teorioista. Tutkielman avulla DataOpsin periaatteita noudattavan dataputken tyypillisimmiksi ominaisuuksiksi koettiin toistettavuus, automatisoitavuus, testattavuus sekä seurattavuus. Ominaisuuksista muodostui tutkimuksen artefakti, jonka avulla onnistuttiin kehittämään DataOpsin käytänteitä osittain mukaileva dataputki. Tämän työn suurimpana kontribuutiona voidaan pitää DataOpsin yleisen tietoisuuden lisäämistä, sekä kuinka sen sisältämiä parhaita käytänteitä voidaan jalkauttaa dataputkien kehitystyöhön teknologista näkökulmaa painottaen.

AVAINSANAT: DataOps, Dataputki, Suunnittelutieteellinen tutkimus, DSRM-malli

Sisällys

1	Johdanto	6
1.1	Tutkimuksen tavoite ja rajaus	7
1.2	Tutkimusmenetelmät	7
2	DataOps	9
2.1	Data-analyysin ongelmat	9
2.2	DataOps määritelmänä	13
2.3	DataOps implementointi	21
2.3.1	Rakennus	22
2.3.2	Suoritus	25
2.3.3	Käyttö	28
2.3.4	Kehitys- ja julkaisuprosessi	29
3	Suunnittelutieteellinen tutkimusprosessi	31
3.1	Tietopohja	32
3.2	Ympäristö	32
3.3	DSRM-prosessimalli	33
3.3.1	Ongelman määrittäminen ja motivointi	33
3.3.2	Tavoitteiden asettaminen	34
3.3.3	Artefaktin suunnittelu	34
3.3.4	Demonstrointi	34
3.3.5	Arviointi	35
3.3.6	Viestintä	36
4	Dataputken kehitysprosessi	37
4.1	Tietopohja	37
4.2	Ympäristö	37
4.3	Ongelman määrittäminen ja motivointi	38
4.4	Tavoitteiden asettaminen	39
4.5	Artefaktin suunnittelu ja kehitys	40

4.6	Demonstrointi	44
4.7	Arviointi	50
4.8	Viestintä	55
5	Diskussio	56
5.1	Tulokset ja johtopäätökset	56
5.2	Rajoitteet ja tutkimuksen arviointi	60
5.3	Jatkotutkimusaiheet	63
	Lähteet	65
	Liitteet	69
	Liite 1. DataOps manifeston kahdeksantoista periaatetta.	69
	Liite 2. Kuvankaappaus Azure Data Factoryn dataputkesta.	72
	Liite 3. Kuvankaappaus onnistuneesta jatkuvan integroinnin sekä toimituksen ajosta Azure Pipelines palvelussa.	73

Kuviot

Kuvio 1.	DataOpsin kolme ulottuvuutta.	17
Kuvio 2.	Alustava viitekehys DataOpsin tutkimukselle.	20
Kuvio 3.	Konseptuaalinen malli dataputkelle.	23
Kuvio 4.	Arvo- ja innovaatioputken prosessi uusien analyysien tuottamiseksi.	27
Kuvio 5.	DataOps kehitys- ja julkaisuprosessi.	29
Kuvio 6.	Suunnittelutieteen tutkimusmalli eli DSRM-malli.	33
Kuvio 7.	Suunniteltu dataputki ajettavaksi kehitys- ja testausympäristössä Azuren pilvipalvelussa.	47
Kuvio 8.	Suunnitellun dataputken jatkuva integraatio sekä toimitus kehitysympäristöstä testausympäristöön.	49

Taulukot

Taulukko 1.	Tyypillisimmät ominaisuudet DataOpsin periaatteita noudattavalle dataputkelle ja sen ajoympäristölle.	43
Taulukko 2.	Yhteenveto demonstroidun dataputken soveltuvuudesta Benghiatin ja muiden listaamiin data-analytiikan haasteisiin.	52
Taulukko 3.	Tutkielman arviointi suunnittelutieteellisen tutkimuksen ohjeistukseen verraten.	61

1 Johdanto

Datan rooli yritysten yhtenä merkittävimmistä kilpailutekijöistä on kasvanut räjähdysmäisesti tällä vuosituhanella. Tämä johtuu osittain siitä, että dataa on tarjolla enemmän kuin koskaan aiemmin useammassa eri muodossa. Lisäksi tiedon määrän odotetaan vain kasvavan lähitulevaisuudessa, kun yhä useampia laitteita kytketään tietoverkkoihin. Esimerkiksi IDC:n (Gantz ja muut, 2018, s. 3) julkaisemassa raportissa datan määrän uskotaan nousevan noin 530 prosenttia vuosien 2018 ja 2025 välillä, jolloin sen kokonaismäärä saavuttaisi maailmanlaajuisesti 175 tsettatavun rajan.

Nopeasti kasvava datan määrä vaatii myös nopeasti tuotettavia ja muokattavia analyysijä, jotta jatkuvasti muuttuvaan kilpailuympäristöön voidaan vastata. Erilaisten trendien sekä asiakkaiden vaihtelevien mieltymysten hyödyntäminen liiketoiminnassa tarvitsee tuekseen toimivaa analytiikkaa, joka reagoi tai parhaimmassa tapauksessa ennakoitulevia muutoksia. Panostaminen yrityksen analytiikkaprosesseihin näkyy esimerkiksi NewVantage Partnersin (NewVantage Partners, 2019, s. 8–9, 11) tekemässä kyselyssä, johon osallistuneista kuudestakymmenestäviidestä Fortune 1000 ryhmään kuuluvasta yrityksestä 91,6 prosenttia koki tarpeelliseksi kiihdyttää investointejaan Big Dataan sekä tekoälyyn liittyen. Lisäksi samasta raportista selviää, että 87,8 prosenttia näistä organisaatioista kokee painetta kiirehtiä investointiensa kanssa. Suurimpana motivaationa tähän pidettiin liiketoiminnan muuttamista sekä parempaa ketteryyttä kilpailukyvyyn kehittäjäksi. Valitettavasti kysely paljastaa myös merkittävän ongelmakohdan rahallisiin panostuksiin liittyen: kuinka nämä strategiset investoinnit onnistutaan jalkauttamaan liiketoimintaa aidosti hyödyttäväksi arvon lisäykseksi. Peräti 95 prosenttia tutkimuksen yrityksistä näki kulttuurin sekä organisationaaliset ongelmat suurimpana esteenä arvon tuottamiselle. Samankaltaisia haasteita on nähtävissä myös PricewaterhouseCoopersin (PwC, 2015) tuottamassa raportissa, jonka mukaan suurimmalta osalta tutkimukseen osallistuneilta yrityksiltä puuttui tarvittavaa osaamista, teknillistä valmiutta sekä oikeinlaista kulttuuria datan hyödyntämiseen.

Näiden dataan kohdistuvien haasteiden ratkaisemiseksi on hiljattain alettu puhumaan termistä DataOps, jonka juuret ulottuvat vuoteen 2014 (Liebmann, 2014). Määritelmänä DataOps pitää sisällään joukon erilaisia parhaita käytänteitä, joiden avulla parannetaan yhteistyötä organisaation sisällä sekä nopeutetaan datan ja siitä johdettavan tiedon jatkuvaa hyödyntämistä.

1.1 Tutkimuksen tavoite ja rajaus

Tutkimuksen tavoitteena on selvittää mitä DataOps määritelmällä tarkoitetaan, ja kuinka sen periaatteita sekä siihen liitettyjä tyypillisimpiä ominaisuuspiirteitä voidaan hyödyntää datan siirtämisen, muokkaamisen ja lataamisen prosessissa, joka tunnetaan laajemmin nimellä dataputki. Terminä DataOps on suhteellisen uusi, sisältäen laajan joukon erilaisia parhaita käytänteitä niin teknologiaan kuin myös organisaation kulttuuriin liittyen. Tässä tutkielmassa keskitytään erityisesti teknologiseen näkökulmaan, tarkemmin data-analytiikassa hyödynnettävään datan siirtämiseen, muokkaamiseen ja lataamiseen liittyviin ohjeistuksiin. Tutkimuksen lopputuloksena saadaan näkemys DataOpsin määritelmästä sekä listaus siihen liitettävistä tyypillisimmistä ominaisuuspiirteistä, jotka huomioiden onnistutaan suunnittelemaan DataOpsin parhaita käytänteitä mukaileva datan siirtämiseen, muokkaamiseen ja lataamiseen kohdistettu prosessi eli dataputki.

Tutkimuksella vastataan seuraavaan tutkimuskysymykseen:

- Millainen on DataOpsin periaatteita noudattava datan siirtämiseen, muokkaamiseen ja lataamiseen kohdistuva prosessi?

1.2 Tutkimusmenetelmät

Tämän tutkielman teoreettinen viitekehys muodostetaan kirjallisuudesta löytyvien DataOpsin määritelmien sekä siihen läheisesti liittyvien käsitteiden pohjalle. Viitekehykseen perustuen suunnittelutieteellisen tutkimuksen avulla luodaan artefakti eli ylätason

listaus tyypillisimmistä ominaisuuksista, jotka huomioimalla pystytään suunnittelemaan DataOpsin periaatteita noudattava datan siirtämisen, muokkaamisen ja lataamisen prosessi eli dataputki. Tutkimuksen rakenne noudattelee suunnittelutieteellisen tutkimuksen DSRM-mallia, jonka avulla suunnitellun artefaktin käyttöä simuloidaan tutkielman demonstraatiovaiheessa. Suunnittelutieteellisen tutkimuksen DSRM-malli on kuvattu tämän työn kappaleessa 3.

2 DataOps

Tässä luvussa tutustutaan data-analyysin tuottamisen haasteisiin, DataOpsin määrittelyyn sekä siihen, kuinka datan siirtämiseen, muokkaamiseen ja lataamiseen kehitetty dataputki onnistutaan implementoimaan DataOpsin periaatteita noudattavaksi.

2.1 Data-analyysin ongelmat

Nykypäivän organisaatiot ovat pakotettuja tekemään tietoisia dataan perustuvia päätöksiä. Osaltaan tämä on seurausta jatkuvasti muuttuvasta kilpailuympäristöstä, jossa nopeaan päätöksentekoon pystyvä yritys voi saavuttaa merkittävää kilpailuetua (Stodder, 2019). Hyvänä esimerkkinä Stodder (2019) kuvaa tilanteen, jossa jälleenmyyjä toimivan data-analyysin avulla pystyy selvittämään trendin asiakkaan mieltymyksissä ennen muita kilpailijoita. Parhaimmassa tapauksessa tämä voi johtaa suurempaan markkinaosuuteen, asiakasuskollisuuteen tai kannattavuuteen. Toisena esimerkkinä voidaan nähdä ennakkoiva analyysimalli, jonka avulla vakuutusyhtiö tai valtion terveysvirasto onnistuu tunnistamaan petosyrityksen ennen kuin se kerkeää aiheuttamaan merkittäviä kustannuksia ja imago tappioita (Stodder, 2019). Tämänkaltaisilla nopeilla ja toimivilla data-analyyseillä pystytään saavuttamaan kilpailuetua, joka voi liittyä niin kannattavuuden parantamiseen, kustannusten laskemiseen kuin myös tuottavuuden nostamiseen. Näihin mahdollisuuksiin on kiinnitetty huomiota myös yritysten strategisella tasolla, jossa investointipäätökset koskien organisaation data- sekä analytiikkaprosesseja ovat rahallisesti kasvaneet huomattavasti (ks. NewVantage Partners, 2019). Valitettavasti kaikesta kiinnostuksesta ja lisätyistä investoinneista huolimatta, perusteellinen datan hyödyntäminen ja muuntaminen oikeasti lisäarvoa tuottavaksi liiketoiminnaksi koetaan suhteellisen matalaksi monessa eri yrityksessä. Esimerkiksi tarkasteltaessa PricewaterhouseCoopersin (PwC, 2015) julkaisemaa tutkimusta, jossa selvitettiin yritysten valmiuksia saavuttaa tietotietua sekä siitä saatavia hyötyjä, peräti 36 prosenttia tutkimuksen kyselyyn osallistuneista 1800 organisaatiosta sijoittui alimpaan kategoriaan ”tietämätön ja huonosti valmistautunut”. Ainoastaan 4 prosenttia kyselyn yrityksistä saavutti ylimmän kategorian ”tietoeliitti”, jossa

yrittäjien valmiuksia datan hyödyntämiseen pidettiin erinomaisina ja edelläkävijöinä omassa sarjassaan.

Miksi organisaatiot eivät siis onnistu jalostamaan hallitsemaansa dataa lisäarvoa tuottavaksi liiketoiminnaksi? Benghiat ja muut (2019, s. 8–12) näkevät osasyllisiksi kahdeksan erilaista kategoriala, miksi data-analyysin tuottaminen aiheuttaa isoja ongelmia eri puolilla organisaatiota. Ensimmäisenä tekijänä he nostavat esille jatkuvasti muuttuvat vaatimukset analyyseille. Syitä tähän voi olla monia sillä usein valmiiden analyyseiden loppukäyttäjät eivät ole asiantuntijoita datan suhteen, jonka vuoksi he eivät välttämättä itse tiedosta mitä tietoa he oikeasti haluavat. Varsinkin liiketoiminnan edustajien on haasteellista tietää ennalta mitä dataa he tulevat tarvitsemaan seuraavien viikkojen aikana. Benghiat ja muut (2019, s. 9) jatkavat toteamalla, että tämä on seurausta nopeatempoisesta kilpailuympäristöstä, jossa mahdollisuudet tulisi hyödyntää heti kun ne avautuvat. Tällöin ei riitä, että analyysi on saatavilla viikkojen viivellä sillä havaittu kilpailuetu voidaan menettää hetkessä ympäristön etsiessä jo vaihtoehtoisia ratkaisuja (Benghiat ja muut, 2019, s. 9). Nopean reagoinnin lisäksi tuotettu analyysi voi paljastaa liudan muita kysymyksiä, joita ei ennalta pystytty määrittelemään (Benghiat ja muut, 2019, s. 9). Useasti loppukäyttäjä haluaa tutkia valmista analyysiä myös toisesta näkökulmasta, jolloin aikaa kuluu uudestaan saman analyyseiden muokkaamiseen uusien vaatimusten mukaiseen muotoon (Lennerholt ja muut, 2018, s. 5056). Tämä vastaavasti synnyttää uusia ajatuksia sekä ideoita dataan ja analyysiin liittyen, joka luonnollisesti lisää entisestään kysyntää uusille vaatimuksille (Benghiat ja muut, 2019, s. 9).

Toisena ongelmana Benghiat ja muut (2019, s. 8–12) esittävät datan elämisen siiloissa. Tällä tarkoitetaan, että data on tallennettuna eri järjestelmiin, joiden välillä ei välttämättä ole yhteyttä eikä riippuvuutta. Osaltaan tähän on vaikuttanut suuresti datan kasvava määrä sekä uusien laitteiden jatkuva lisääminen eri tietoverkkoihin (ks. Gantz ja muut, 2018). Erilaiset järjestelmät rakennetaan useilla teknologioilla, joka vaatii data-analytiikan tiimeiltä perehtymistä uuteen järjestelmään (Benghiat ja muut, 2019, s. 10). Perehtyminen voi sisältää esimerkiksi muutoksia järjestelmän arkkitehtuuriin,

pääsyoikeuksien hallinnoimista tai testausta uusien analytiikkaprosessien implementoinnin yhteydessä (Benghiat ja muut, 2019, s. 10). Muutoksien läpivieminen voi olla monimutkaista, joka hidastaa prosessien etenemistä, mikä puolestaan synnyttää pullonkauloja analytiikan muodostamiselle (Benghiat ja muut, 2019, s. 10). Lisäksi tietojärjestelmien tietoturvaan liittyvät tekijät, kuten kysymykset koskien turvallista datan siirtoa eri järjestelmien välillä, voivat aiheuttaa viivästyksiä datan saatavuuteen (Balachandran & Prasad, 2017, s. 1119–1120; Merchan, 2018).

Kolmantena listauksen kategoriana Benghiat ja muut (2019, s. 10) pitävät datan muotoa operationaalisissa järjestelmissä. Hyvänä esimerkkinä voidaan pitää yksinkertaista verkkosivustoa, jonka kautta asiakkaan on mahdollista ostaa erilaisia tuotteita. Tämänkaltaisen sovelluksen tietokanta-arkkitehtuuri suunnitellaan usein tukemaan tosiaikaista transaktiokäsittelyä (englanniksi Online Transaction Processing, OLTP), jolloin ideana on tallentaa, päivittää tai noutaa vain pieni määrä tietueita tietokannasta sen avaimien ja indeksien avulla (Kleppmann, 2017, s. 90; Chaudhuri & Dayal, 1997, s. 65). Data-analytiikan kohdalla tämä periaate aiheuttaa ristiriidan, sillä analyyseja varten luodut tietokantakyselyt tarvitsevat usein valtavan määrän tietokannan tietueita, jotta erilaiset tilastotieteelliset menetelmät, kuten yksinkertaiset summaus tai keskiarvon määrittäminen, pystytään laskemaan analyysin avulla tuotettuja raportteja varten (Kleppmann, 2017, s. 90–91; Chaudhuri & Dayal, 1997, s. 65). Tämä osin vastakkainen lähestymistapa tunnetaan myös nimellä tosiaikainen tiedon jalostus (englanniksi Online Analytical Processing, OLAP) (Kleppmann, 2017, s. 91; Chaudhuri & Dayal, 1997, s. 65).

Neljäntenä data-analytiikan ongelmana Benghiat ja muut (2019, s. 10–11) näkevät datavirheisiin liittyvät haasteet. Mikään järjestelmä ei ole täydellinen, ja siksi jokainen yritys tulee kohtaamaan datan laatuun liittyviä haasteita. Benghiat ja muut (2019, s. 10–11) luettelevat esimerkeiksi dataputken rikkoutumisen dataan kohdistuvan virheen vuoksi, duplikaatit tietueissa, virheellisen datan yksittäisessä kolumnissa, uuden algoritmin odotuksenvastaisen toiminnan, tietokanta-arkkitehtuurin muuttumisen tai virhetilan tietojärjestelmässä. Näihin virheisiin liittyvät ongelmat voivat pahimmillaan

johtaa huonolaatuisiin raportteihin, jotka johtavat viidenteen ongelmakohtaan eli virheelliseen raportointiin Benghiatin ja muiden (2019, s. 11) listassa. Virheellinen raportointi lisää tyytymättömyyttä loppukäyttäjien keskuudessa, sillä raportoinnin korjaaminen vaatii aina lisää aikaa analyysin tekijältä, joka vastaavasti hidastaa uusien raporttien tuottamista. Benghiat ja muut (2019, s. 11) jatkavat toteamalla, että jatkuva virheellinen raportointi voi vaikuttaa analytiikkatiimin uskottavuuteen, jolloin loppukäyttäjät alkavat epäilemään analyysien luotettavuutta tai niiden tuomaa arvoa.

Data-analytiikan prosessia voidaan kuvata putkena, jossa tietty määrä prosesseja pyrkii tuottamaan johdonmukaisen lopputuloksen (Benghiat ja muut, 2019, s. 11). Samoin kuin datavirheiden kohdalla, myös dataputkeen voi tulla ongelmia, kun esimerkiksi datan sisältämää lähdejärjestelmää muutetaan tai analytiikkaa muokataan vastaamaan paremmin uusiin vaatimuksiin (Benghiat ja muut, 2019, s. 11). Benghiat ja muut (2019, s. 11) pitävät edellä esitettyä dataputken jatkuvaa päivittämistarvetta kuudentena analytiikan kohtaamana haasteena. Muutoksien tekeminen dataputkeen voi viedä huomattavasti aikaa, sillä näiden muutoksien yhteydessä joudutaan usein varmistamaan sekä testaamaan jo aikaisemmin tuotettujen analyysien toimivuus (Benghiat ja muut, 2019, s. 11). Benghiat ja muut (2019, s. 11) esittävät, että suurin osa analytiikkojen, datatieteilijöiden sekä datainsinöörien ajasta kuluu juuri analytiikkaprosessin dataputkien päivittämiseen ja toimivuuden varmistamiseen. Tällainen loppukäyttäjiltä näkymättömissä kulutettava aika vie luonnollisesti resursseja uusien analyysien tuottamiselta, joka osaltaan hidastaa analyysiprosessin toimivuutta.

Seitsemäntenä haasteena esitetään analyysin tuottamiseen rakennetut manuaaliset prosessit (Benghiat ja muut, 2019, s. 11). Valitettavan usein raporttien tuottamiseen yrityksissä käytetään valtava määrä manuaalista työtä ja vain pieni osa analyysiprosessista on automatisoitu (Benghiat ja muut, 2019, s. 11). Manuaalinen työ on aikaa vievää sekä virhealtista, joka osaltaan voi nostaa työn valmistumiseen käytettyjen työtuntien määrää merkittävästi (Benghiat ja muut, 2019, s. 11). Käsintehdyt suoritukset liittyvät läheisesti Benghiatin ja muiden (2019, s. 11–12) viimeiseen

analytiikan kohtaamaan haasteeseen eli ”toivon ja sankaruuden ansaan”. Jatkuvat muutokset, peruutetut projektit, vaihtuvat aikataulut, tyytymättömät loppukäyttäjät, joustamattomuus, huono laatu, analyysin alhainen tuotto suhteessa käytettyihin resursseihin sekä tarpeettomat ominaisuudet ovat tekijöitä, jotka pakottavat raporttien tekijöitä turvautumaan juuri edellä mainittuihin manuaalisiin työvaiheisiin (Benghiat ja muut, 2019, s. 11–12). Tämä voi osaltaan johtaa ylipitkiin työpäiviin, jotta asetettuihin odotuksiin pystytään vastaamaan ajallaan (Benghiat ja muut, 2019, s. 12). Valitettavasti nämä manuaaliset prosessit aiheuttavat ongelmia yrityksen koon kasvaessa. Lisäksi Benghiat ja muut (2019, s. 12) näkevät testauksen jäävän usein taka-alalle, kun aikatauluista halutaan pitää kiinni. Tämä on omiaan nostamaan analyysiin kohdistuvia riskejä sekä datan laadun että prosessin toimivuuden suhteen. Testaamisella pystytään parantamaan analytiikan laatua mutta esimerkiksi uusien vaatimusten integroiminen osaksi analyysia voi olla haastavaa, ja vaatii usein uudelleen testaamista. Tämä taas johtaa hitaampaan tuottavuuteen sekä mahdollisesti tyytymättömyyteen loppukäyttäjien keskuudessa (Benghiat ja muut, 2019, s. 11–12). Toisaalta liian kattava testaaminen voi myös aiheuttaa haasteita, sillä testaus kuluttaa resursseja uusien toiminnallisuuksien kehitystyöltä.

Kuten Benghiatin ja muiden (2019, s. 9–12) listauksesta nähdään, data-analyysin tuottamiseen kohdistuu monia erilaisia haasteita. Osa niistä liittyy enemmän teknologisiin tekijöihin, osa vastaavasti korostaa vahvemmin organisaation kulttuuriin liitettäviä ongelmatapauksia. Näiden haasteiden ratkaisemiseksi Lenny Liebmann (2014) esitti termin nimeltä DataOps, jonka tarkempaan määritelmään tutustutaan seuraavaksi.

2.2 DataOps määritelmänä

Vuonna 2014 Lenny Liebmann (2014) esitti ajatuksen parhaista käytänteistä, joiden tavoitteena olisi parantaa kommunikaatiota datatieteilijöiden sekä yrityksen tuotannon prosessien välillä. Hänen mukaansa tuotettujen analyysimallien siirtäminen nopeasti tuotantokäyttöön koki vastaavanlaisia ongelmia kuin mitä ohjelmistokehityksen

lopputuotteena saavutetut ohjelmistopäivitykset: kuinka siirtää kehityksen aikana uudistettu ohjelmistokoodi nopeasti tuotantoon ja samalla varmistaa sen toimivuus? Ohjelmistokehityksen kohdalla ratkaisuksi kehitettiin joukko käytänteitä, joiden tarkoituksena oli parantaa yhteistyötä juuri ohjelmistokehityksen ja tuotannon prosessien välillä (Liebmann, 2014). Näiden käytänteiden pohjalta muodostui lopulta termi DevOps, jossa organisatorisella lähestymistavalla pyritään kiihdyttämään muutoksien tuottamista sekä operointia joustavissa järjestelmissä, korostaen eri toiminnallisten ryhmien välistä yhteistyötä etenkin ohjelmistokehityksen sekä tuotannon toimintojen kesken (Dyck ja muut, 2015). Liebmann (2014) näkee samankaltaisen yhteistyön olevan suuressa roolissa DataOpsin kohdalla, tosin kommunikaatiota korostetaan enemmän datatieteilijöiden ja tuotannon toimintojen välillä, DevOpsin painottaessa ohjelmistokehityksen näkökulmaa. Liebmanin (2014) mukaan koordinaation parantamista näiden sidosryhmien välillä voidaan pitää erittäin kriittisenä tekijänä IT-organisaation menestymismahdollisuuksille valitsevan reaaliaikaisen data-analytiikan maailmassa.

Myös Palmer (2015) näkee samankaltaisen yhteistyön olevan DataOpsin lähtökohtana, tosin hän laajentaa termin kattamaan myös enemmän teknologista näkökulmaa. Palmer (2015) kuvailee DataOpsia datanhallinnan menetelmäksi, jossa korostetaan data-ammatilaisten kykyä kommunikaatioon, integraatioon, automatisointiin, yhteistyöhön sekä sen mitattavuuteen. Hänen mukaansa määritelmän tavoitteena on auttaa organisaatiota tuottamaan nopeasti dataa, kiihdyttämään analytiikan tuottamista sekä mahdollistamaan aikaisemmin mahdottomien analyysien kehittämistä. Palmer (2015) katsoo tavoitteen olevan mahdollinen, kun DataOps nähdään yhdistelmänä dataan liittyvää käsittelyä, integraatiota, laatua sekä suojausta ja yksityisyyttä painottavana terminä.

Teknologista määrittelyä jatkaa myös Ereth (2018, s. 5) omassa ehdotuksessaan, jossa hän katsoo DataOpsin olevan enemmän joukko erilaisia käytäntöjä sekä teknologioita, eikä sinällään varsinainen metodi tai työkalu. Tutkimuksessaan hän ehdottaa määritelmäksi seuraavaa kuvausta:

DataOps on joukko käytänteitä, prosesseja ja teknologioita, joissa yhdistyvät integroitu ja prosessikeskeinen datanäkökulma ketterän ohjelmistokehityksen automaation ja menetelmien avulla, tavoitteena laadun, nopeuden ja yhteistyön parantaminen sekä jatkuvan kehittymisen kulttuurin korostaminen.

Määritelmän voidaan katsoa mukailevan hyvin pitkälle Palmerin (2015) kuvausta, sillä molemmissa painottuvat automatisointi, integraatio sekä yhteistyö datanhallinnan välineenä. Lisäksi molemmat näkevät analytiikan nopeuden lisääntymisen yhdeksi määritelmän tavoitteista. Erethin kuvaus on linjassa myös Liebmannin (2014) alkuperäisen ajatuksen kanssa, mainiten yhteistyön parantamisen toisena tavoitteena. Keskeisimpänä liisäyksenä Ereth yhdistää ketterän ohjelmistokehityksen sekä jatkuvan kehittymisen suoraan osaksi DataOpsin määritelmää. Yksinkertaistaen ketterällä ohjelmistokehityksellä tarkoitetaan kehitystyötä, jossa tavoitteena on luoda lisäarvoa tuottamalla toimivaa ohjelmistoa säännöllisesti lyhyiden toimintasykliä avulla (Dingsøyr ja muut, 2012, s. 1214; Beck ja muut, 2001). Ketterään kehittämiseen liittyy valtava joukko erilaisia käytänteitä, joita hyödyntämällä ohjelmiston toiminnallisuudesta saadaan jatkuvaa palautetta, mihin itseohjautuvat kehitystiimit pystyvät tarvittaessa reagoimaan kehitysprosessin jokaisessa vaiheessa (Dingsøyr ja muut, 2012, s. 1214; Beck ja muut, 2001).

Samankaltaista ajattelua prosessien ja teknologioiden keskinäisestä toimivuudesta jatkavat Sahoo ja muut (2019, s. 1–2). Heidän ajatuksensa mukaisesti DataOps on prosessikeskeinen metodologia, jota datan parissa työskentelevät henkilöt hyödyntävät parantaakseen data-analyysien laatua ja kiertonopeutta, tavoitteenaan edesauttaa parempien liiketoimintapäätösten tekemistä sekä tuottojen kasvattamista, tuoden samalla dataa organisaation hyödynnettäväksi. Tämän he katsovat mahdolliseksi, kun DevOpsin määritelmän pohjalta perityt automatisoidut, toistettavat sekä täsmälliset prosessit liitetään osaksi DataOpsia. Käytänteiden avulla kehitetyt datan siirtämisen ja muokkaamisen prosessit eli dataputket pystyvät tuottamaan jatkuvaa lisäarvoa yrityksessä samalla vähentäen datan siiloutumista eri järjestelmien kesken. Sahoo ja muut (2019, s. 1–2) näkevät DataOpsin olevan suora johdannainen DevOpsin konseptista. Sama ajatus on nähtävissä myös Capizzin ja muiden (2019, s. 1–5) päätelmässä, jonka mukaan DataOps pyrkii kehittämään data-analytiikan elinkaarta hyödyntämällä DevOpsin mukaisia sääntöjä sekä

periaatteita, joihin lukeutuvat esimerkiksi kommunikaatio eri sidosryhmien kesken, yhteistyö, automaatio sekä integraatio. Tutkimuksessaan he painottavat eri teknologisten työkalujen ja prosessien tuomia hyötyjä, jotka ovat edistäneet ohjelmistokehityksen ja tuotannon keskinäistä toimivuutta, ja jotka pohjautuvat juuri DevOpsin mukaiseen ajatteluun.

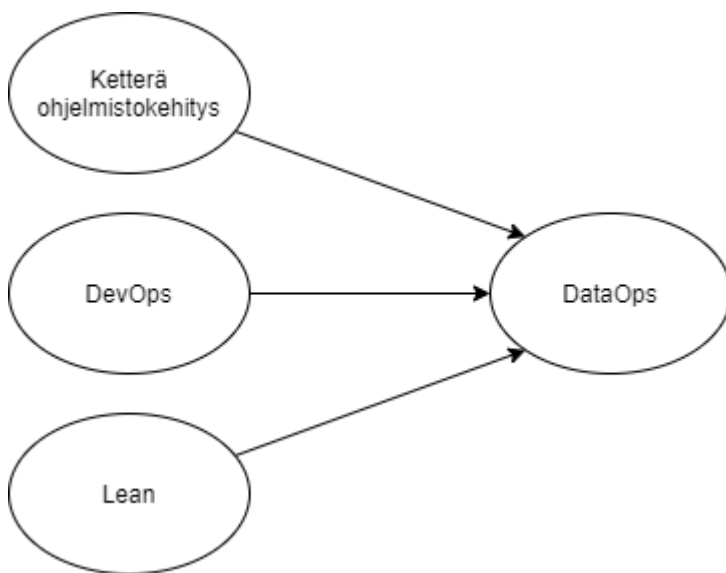
Vaikka DevOpsin roolia painotetaan vahvasti kirjallisuudesta löytyvien DataOpsin tämänhetkisten määritelmien kohdalla, voidaan se myös nähdä vain yhtenä, tosin tärkeänä osana laajempaa määritelmää. Esimerkiksi Munappy ja muut (2020, s. 169) havaitsivat tutkimuksessaan eri käyttäjien määrittelevän käsitteen eri painopisteiden kautta: osa lähestyi termiä enemmän siihen sisältyvien aktiviteettien tai teknologioiden kautta, osa vastaavasi koki lopputavoitteiden tai organisaation sidosryhmien rakenteen olevan määrittelevämpi kuvaaja. Munappy ja muut (2020, s. 169) näkevät DataOpsin olevan oma erillinen kokonaisuutensa, joka tosin jakaa samoja piirteitä DevOpsin sekä ketterän ohjelmistokehityksen kanssa. Suurimpana erona voidaan nähdä datan rooli: DevOpsin ja ketterän ohjelmistokehityksen keskittyessä enemmän ohjelmistokoodin toimivuuteen, DataOps painottaa tämän lisäksi itse dataa prosessikeskeisessä lähestymistavassa (Munappy ja muut, 2020, s. 169). DataOps ei siis suoraan kopioi parhaita käytänteitä tai työkaluja edellä mainituista ajatuksista, vaan nämä toimiviksi koetut prosessit joudutaan räätälöimään ja sovittamaan data-analytiikan käyttöön juuri sen datakeskeisen eroavaisuuden vuoksi. Tämä ajatus voidaan nähdä Munappyn ja muiden (2020, s. 169) ehdottamassa määritelmässä:

DataOps voidaan määritellä lähestymistapana, joka kiihdyttää korkealaatuisten tuloksien toimitusta automatisoimalla ja orkestroimalla datan elinkaaren vaiheita. DataOps ottaa käyttöön parhaimmat käytänteet, prosessit, työkalut ja teknologiat ketterästä ohjelmistokehityksestä sekä DevOpsista hallitsemaan analytiikan kehittämistä, optimoimaan koodin tarkastusta, rakentamaan ja toimittamaan uusia analyysijä, siten edistämällä yhteistyön kulttuuria ja jatkuvaa kehittymistä.

Munappyn ja muiden määritelmä muistuttaa vahvasti Erethin (2018, s. 5) kuvausta, sillä myös hän mainitsee määrittelyssään prosessien, käytänteiden ja teknologioiden roolit keskeisenä datan laadun sekä nopeuden kehittämisessä. Lisäksi molemmat määritelmät

edellyttävät yhteistyötä sekä jatkuvan kehittymisen kulttuuria. Munappy ja muut (2020, s. 169) tosin kokevat määritelmänsä DataOpsista dataan liittyväksi strategiaksi, sillä se luo pohjan muutokselle. He näkevät dataan rinnastettavan strategian olevan edellytyksenä kaikille organisaatioille, jotka hyödyntävät dataa analytiikan parissa.

Myös Benghiat ja muut (2019, s. 22–23) näkevät DataOpsin olevan enemmän kuin pelkkä lainaus DevOpsin parhaista käytänteistä. Heidän mukaansa määritelmä voidaan nähdä yhdistelmänä jo aikaisemmin mainittuja ketterää ohjelmistokehitystä ja DevOpsia mutta osittain myös tilastollista prosessikontrollia (englanniksi Statistical Process Control, SPC). Benghiat ja muut (2019, s. 22–23) ehdottavat termin muodostuvan näiden kolmen eri ajatuksen pohjalta alla olevan kuvion 1 mukaisesti:



Kuvio 1. DataOpsin kolme ulottuvuutta (mukaillen Benghiat ja muut, 2019, s. 22).

Kuvion 1 mukaisesti, Benghiat ja muut (2019, s. 17–23) näkevät DataOpsin muodostuneen ketterän ohjelmistokehityksen, DevOpsin sekä lean-ajattelun pohjalta, lainaten parhaita käytänteitä sekä prosesseja kaikilta kolmelta eri alueelta. Heidän näkemyksensä tuo uutena ominaisuutena lean-ajattelun, tarkemmin eriteltynä tilastollisen prosessikontrollin osaksi DataOpsin määritelmää. Lean-ajattelu on laaja käsite mutta yksinkertaistaen se pohjautuu suurelta osin toimintamalliin, jonka ideana on optimoida

organisaation prosesseja esimerkiksi toimittamalla tuotteita vasta tarvittaessa Just-In-Time menetelmän mukaisesti, vähentämällä resursseja, kehittämällä parannuksia strategioiden avulla, hallitsemalla virhetilanteita, standardoimalla menetelmiä tai ottamalla käyttöön erilaisia tieteellisiä tekniikoita (Pettersen, 2009). Näiden prosessien toimintaa pystytään vastaavasti seuraamaan tilastollisen prosessikontrollin avulla, joka on yhdistelmä tilastotieteellisiä tekniikoita prosessien toimintakyvyn monitoroimiseen (ASQ, 2021). Benghiat ja muut (2019, s. 21–22) katsovat näiden tekniikoiden olevan tärkeä osa datan laadun varmistusta, etenkin kun tarkastellaan jo tuotantokäytössä olevia datan siirtämiseen ja muokkaamiseen tarkoitettuja dataputkia sekä prosesseja.

Kolmen ulottuvuuden eli ketterän ohjelmistokehityksen, DevOpsin sekä lean-ajattelun yhdistelmä on havaittavissa myös DataOpsin ensimmäisissä kirjoitetuissa periaatteissa, jotka yhdessä muodostavat yhtenäisen julistuksen nimeltään DataOps manifesto (DataOps Manifesto, 2021). Tämän manifeston mukaan analytiikalle onnistutaan tuottamaan eniten arvoa, kun datatyön painopisteitä kohdistetaan enemmän kohti seuraavia näkökulmia:

1. Yksilöitä ja kanssakäymistä prosessien sekä välineiden sijasta
2. Toimivaa analytiikkaa kattavan dokumentaation sijasta
3. Asiakasyhteistyötä sopimusneuvottelujen sijasta
4. Kokeilua, iterointia ja palautetta laajamittaisen etukäteissuunnittelun sijasta
5. Monialaista toimintojen vastuunottoa siiloutuneiden vastuualueiden sijasta

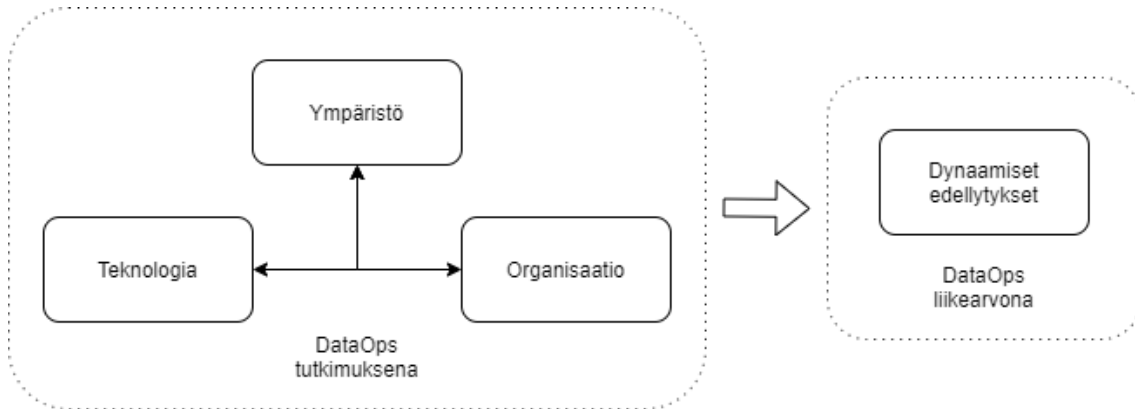
Nämä viisi ajatusta muistuttavat vahvasti ketterän ohjelmistokehityksen kirjallisuudesta löytyvää ketterän ohjelmistokehityksen julistusta, jonka allekirjoittajien mukaan parhaimmat tavat tehdä ohjelmistokehitystä löytyvät painottamalla yksilöitä ja kanssakäymistä, toimivaa ohjelmistoa, asiakasyhteistyötä sekä muutokseen vastaamista (Agile Manifesto, 2021). DataOpsin julistuksen keskeisenä erona on asioiden painottaminen analytiikkaan kohdistuvien ajatuksien kautta, missä ketterä ohjelmistokehitys arvottaa asioita luonnollisesti siihen enemmän liittyvien tekijöiden pohjalta. Lisäksi DataOps manifesto jakaa monialaisten toimintojen vastuunoton omaksi kohdakseen, joka pyrkii

huomioimaan esimerkiksi datan siiloutumisen tuottamat haasteet (ks. Benghiat ja muut, 2019, s. 9–10). Toisaalta on tärkeää huomioida, että myös julistuksen jälkimmäisillä asioilla on arvoa eikä niitä tulisi unohtaa. Arvostamalla kuitenkin enemmän listan ensiksi mainittuja asioita, onnistutaan manifeston allekirjoittaneiden mukaan tuottamaan paremmalla tavalla arvoa data-analytiikassa (DataOps Manifesto, 2021). Sama ajatus nähdään jälleen ketterän ohjelmistokehityksen julistuksessa, jonka mukaan myös jälkimmäisillä asioilla on arvoa, mutta ensiksi mainittuja kohtia arvostetaan enemmän (Agile Manifesto, 2021).

DataOps manifeston viiden ajatuksen pohjalta on julistukseen lisätty yhteensä kahdeksantoista erillistä periaatetta, jotka auttavat kohdistamaan data-analytiikan aihealueita kohti DataOps manifeston mukaista arvoa tuottavaa analytiikkaa (DataOps Manifesto, 2021). Nämä periaatteet ja niihin liitetyt tarkemmat kuvaukset on esitetty tämän tutkielman liitteessä 1, jossa ne ovat vapaasti käännettynä suomenkieliseen muotoon. Julistus kohdistaa huomiota sekä organisatorisiin tekijöihin kuin myös teknisiin yksityiskohtiin, joten sitä voidaan pitää toimivana lähtökohtana ja kokoelmana DataOpsiin liitettäviä käytänteitä. Manifesto ei kuitenkaan tarjoa varsinaista määritelmää DataOpsille mutta sen periaatteita pystytään hyödyntämään myös tämän tutkielman artefaktin luomisen kohdalla, etenkin huomioimalla sen sisältämiä teknisiä ajatuksia.

Kuten edellä olevassa teoriakatsauksessa nähtiin, ei DataOpsin määritelmälle ole vielä löydettävissä yhtä yksiselitteistä kuvausta tieteelliseen kirjallisuuteen pohjautuen. Katsauksen perusteella määritelmä voidaan kuvata painottaen joko enemmän teknologista näkökulmaa tai vastaavasti hyödyntäen voimakkaammin organisaation rakenteeseen ja kulttuuriin pohjautuvia tavoitteita. Huomionarvoista kuitenkin on, etteivät nämä erilaiset painotukset ole toisiaan pois sulkevia tekijöitä. Huolimatta vaihtelevista painopisteistä, jokainen määritelmä yhdistää jollain tasolla teknologiset prosessit sekä organisaatioon liittyvät aihealueet keskenään. Tämä yhdistävä ajatus näkyy Erethin (2018, s. 5–6) tutkimuksessa, jossa hän ehdottaa DataOpsiin liittyvälle tutkimukselle Teknologia-

Organisaatio-Ympäristö viitekehikseen (englanniksi Technology-Organization-Environment Framework, TOE) perustuvaa mallia alla olevan kuvion 2 mukaisesti:



Kuvio 2. Alustava viitekehys DataOpsin tutkimukselle (mukaillen Ereth, 2018, s. 6).

Kuten kuviosta 2 nähdään, DataOpsin tarkastelu voidaan jakaa kahteen pääalueeseen: DataOps tutkimuksena sekä DataOps liikearvona. Tutkimukseen perustuvassa tarkastelussa painotetaan enemmän itse määritelmään liittyviä metodeja, teknologioita sekä siihen perustuvia konkreettisia implementointeja (Ereth, 2018, s. 5). Ereth näkee TOE-viitekehikseen pohjautuvan tutkimuksen olevan toimiva lähtökohta DataOpsille, sillä sen avulla tutkimusta voidaan painottaa joko teknologiaan, organisaatioon tai ympäristöön liittyvien alueiden kautta. Tämä on hyvin linjassa erilaisten määritelmien kanssa, sillä vaikka DataOps nähdään väistämättä näiden kaikkien kolmen yhdistelmänä (vrt. Ereth, 2018; Munappy ja muut, 2020), tarkastelua voidaan lisäksi painottaa joko enemmän teknologiseen alueeseen keskittyen (vrt. Sahoo ja muut, 2019; Capizzi ja muut, 2019) tai vastaavasti organisaatioon (vrt. Liebmann 2014; Munappy ja muut, 2020) liittyvien aihealueiden kautta. DataOpsin tuottamaa lisäarvoa liiketoiminnan näkökulmasta pystytään taas tarkastelemaan dynaamisten edellytysten kautta, jonka Ereth näkee sopivana pohjana DataOpsin avulla saatavan liikearvon tarkasteluun (ks. Teece ja muut, 1998).

Tämän tutkimuksen voidaan nähdä sijoittuvan Erethin esittämän mallin mukaisesti DataOpsin tutkimukseen, missä tavoitteena on luoda listaus tyypillisimmistä DataOpsin ominaisuuksista tarkastellen datan siirtämiseen, muokkaamiseen ja lataamiseen liittyvää

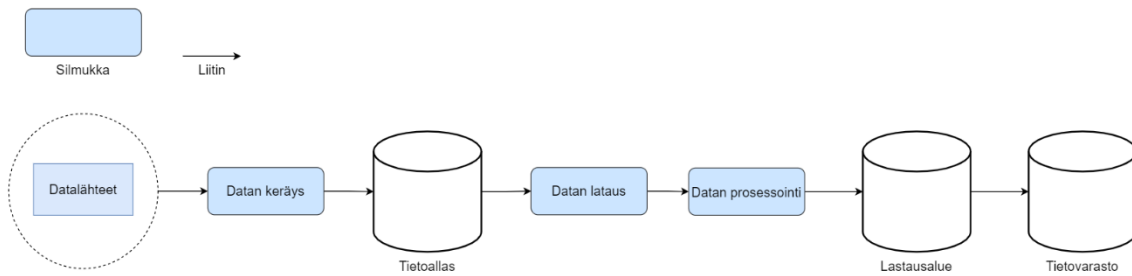
prosessia eli dataputkea. Tutkimuksen painopiste on teknologisessa aihealueessa, jättäen ympäristön ja ihmiset vähemmälle huomiolle. Toisaalta tutkimuksen DSRM-mallin mukaisessa demonstraatiovaiheessa suunnitellaan onnistuneesti DataOpsin piirteitä mukaileva dataputki teoriakatsauksen pohjalta luotua listausta seuraten, joka voi tuottaa lisäarvoa niin ympäristön, ihmisten tai liikearvon suhteen. Aihealueet eivät siis ole toisiinsa poissulkevia tekijöitä, vaan ne voidaan mieltää enemmän tutkimuksen tarkastelua tai lähtökohtia ohjaaviksi tekijöiksi Erethin (2018, s. 5–6) mallin kuvauksen mukaisesti. Samankaltainen ajatus on selkeästi nähtävillä myös suunnittelutieteellisen tutkimuksen DSRM-mallissa, jossa tutkimuksen lähtökohdat voidaan nähdä juuri yhdistelmänä aihealueita liittyen teknologiaan, organisaatioon sekä ihmisiin. Ja juuri tämä havainto toimi myös pohjana tämän tutkimuksen tutkimusmenetelmän valinnalle, joka on esitelty tarkemmin kappaleessa 3. Seuraavaksi tutkitaan, mitä erilaisia vaiheita ja prosesseja DataOpsin implementointi voi sisältää teknologista näkökulmaa painottaen.

2.3 DataOps implementointi

DataOpsin ollessa vielä suhteellisen uusi tieteellinen tutkimuskohde, on sitä hyödyntäviä tutkimukseen perustuvia teknologisia toteutuksia vähän saatavilla. Määritelmän osittainen avoimuus ja yksittäisen yleisesti alalla hyväksytyn kuvauksen puuttuminen johtavat väistämättä moniin mahdollisiin toteutustapoihin. Toisaalta monissa kuvauksissa on havaittavissa toisiaan yhdistäviä sekä täydentäviä ajatuksia, joiden kautta esimerkiksi Capizzi ja muut (2019, s. 4–5) sekä Patterson (2018) näkevät DataOpsiin liitettävän teknologisen implementointivaiheen sisältävän ainakin seuraavat kolme päävaihetta: rakenus (englanniksi Build), suoritus (englanniksi Execute) ja käyttö (englanniksi Operate). Yhdessä nämä vaiheet muodostavat jatkuvan kierteen eli DataOpsin kehitys- ja julkaisuprosessin, jossa vaiheita toistetaan uudelleen samassa järjestyksessä data-analytiikan kehityksessä (Benghiat ja muut, 2019, s. 37).

2.3.1 Rakennus

Rakennusvaiheessa kehittäjät keskittyvät rakentamaan datan siirtämiseen ja muokkaamiseen tarkoitettuja dataputkia, joiden avulla heterogeenisista datalähteistä kerätty data muokataan tarvittavin menetelmin ja siirretään lopuksi kohdepaikkaan varsinaista käyttöä eli data-analytiikkaa varten (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Dataputki voidaan siis nähdä joukkona toisiinsa liittyviä komponentteja, jotka suorittavat datan siirtämiseen, muokkaamiseen tai lataamiseen liittyviä aktiviteetteja, ja joissa edellisen komponentin tuloste on seuraavan komponentin syöte (Raj ja muut, 2020, s. 17–18). Putki muistuttaa siis pitkälti vakiintuneempaa datalatauksiin liittyvää käsitettä ETL-prosessi, joka on lyhenne sanoista siirto (englanniksi Extract), muokkaus (englanniksi Transform) sekä lataus (englanniksi Load). ETL-prosessin tavoite on identtinen dataputken ajatuksen kanssa, sillä myös ETL-prosessissa eri lähdejärjestelmistä saatava lähde-data siirretään kohti yhtenäistä tallennustilaa hyödyntäen dataan liittyviä muokkaus- ja puhdistusmenetelmiä (Kleppmann, 2017, s. 92; Chaudhuri & Dayal, 1997, s. 66–67). ETL-prosessiin sisältyvien aktiviteettien suorituseräjäjärjestystä voidaan toisaalta myös muuttaa, jolloin lähdejärjestelmän data ladataan ensin yhtenäiseen tallennustilaan, ja vasta tämän jälkeen sitä muokataan sopivin menetelmin (Haryono ja muut, 2020, s.116). Tästä periaatteesta käytetään nimitystä ELT-prosessi, jonka isoin ero ETL-prosessiin on edellä esitetty lataamisen ja muokkauksen suorituseräjäjärjestys, joka on varsinkin viime vuosina alkanut yleistymään datan siirtotapana (Haryono ja muut, 2020, s.116–119). Raj ja muut (2020, s. 14) näkevät dataputken olevan terminä laajempi käsite, sisältäen nämä molemmat prosessit. Tätä lähtökohtaa osaltaan hyödyntäen Raj ja muut (2020) suunnittelivat tutkimuksessaan yhtenäisen konseptuaalisen mallin dataputkelle, jota voisi hyödyntää organisaatiossa, joka käsittelee dataan liittyviä applikaatioita. Malli on tarkoitettu vastaamaan myös osittain datanhallinnan ja analytiikan mukanaan tuomiin haasteisiin esimerkiksi datan laadun tai datavirheiden näkökulmasta (vrt. Data-analyysin ongelmat; Raj ja muut, 2020, s. 18–19). Dataputken konseptuaalinen malli on esitetty yksinkertaistetussa muodossa alla olevassa kuviossa 3:



Kuvio 3. Konseptuaalinen malli dataputkelle (mukaihen ja yksinkertaistettu Raj ja muut, 2020, s. 17–18).

Kuvion 3 mallin mukaisesti dataputki muodostuu silmukoista (englanniksi Node), joiden välillä dataa liikutetaan liittimien (englanniksi Connector) avulla. Silmukat voivat sisältää erilaisia dataan liittyviä aktiviteetteja riippuen niiden sisältämisestä ja tarvitsemista ominaisuuksista. Raj ja muut (2020, s. 17–18) näkevät tällaisiksi ominaisuuksiksi datan luomisen, keräämisen, lataamisen, tallentamisen, prosessoinnin, merkinnän, alustavan prosessoinnin sekä vastaanottamisen. Liittimien tehtävänä on vastaavasti yhdistää silmukat toisiinsa sekä siirtää joko raakamuotoista tai käsiteltyä dataa eri aktiviteettien välillä. Liittimiin voi myös liittyä erilaisia ominaisuuksia kuten datan välittäminen sekä autentikointi, validointi, monitorointi, virnehallinta tai hälytysten lähettäminen. Tämänkaltaisesta dataputkesta, jossa data liikkuu yhdensuuntaisesti silmukasta toiseen liittimien kautta, käytetään myös termiä suunnattu asyklinen kaavio (englanniksi Directed Acyclic Graph, DAG). Nimensä mukaisesti suunnatussa asyklisessä kaaviossa silmukka ei ikinä muodosta yhteyttä liittimen avulla aikaisempaan silmukkaan, vaan data liikkuu ennalta määrättyssä järjestyksessä aina seuraavaan silmukkaan (Munappy ja muut, 2020, s. 173).

Mallia mukailevan dataputken ensimmäisessä silmukassa data kerätään eri lähdejärjestelmistä ja tallennetaan alkuperäisessä muodossaan yhtenäiseen tallennuspaikkaan. Lähdejärjestelmät ovat siis datantuottajia, jotka voivat muodostaa toisiinsa verrattuna hyvin erityyppistä ja heterogeenista dataa (Raj ja muut, 2020, s. 17–18). Tietueet voivat olla peräisin koneista tai järjestelmistä, jotka luovat dataa suoraan ilman ihmisen väliintuloa. Toisaalta data voi syntyä myös ihmisen toiminnan lopputuloksena esimerkiksi työntekijöiden aktiviteettien kautta. Raj ja muut (2020, s. 17–18) jakavat edellä mainitut

datantuottajat joko konelähtöisiin tai ihmislähtöisiin datalähteisiin, riippuen siitä kuka tai mikä toimii datan luomisprosessin käynnistäjänä. Nämä lähdejärjestelmät voivat luoda dataa eri aikoina sekä erilaisilla aikajänneillä, jolloin datan kerääminen voi tapahtua joko sarjoina, ajoittaisena tai jatkuvana datan siirtona (Raj ja muut, 2020, s. 18). Kerätty data siirretään ja tallennetaan yhtenäiseen tallennustilaan, säilyttäen samalla sen alkuperäinen muoto (Raj ja muut, 2020, s. 18). Tämänkaltaisesta keskitetystä raakadatan tallennuspaikasta käytetään myös nimeä tietoaallas (englanniksi Data Lake), jonka tarkoituksena on säilyttää kerätty data mahdollisimman lähellä alkuperäistä muotoaan, jotta tiedostojen lukeminen uudelleen olisi tarvittaessa mahdollista myös tulevaisuudessa (Mathis, 2017, s. 289–290; Raj ja muut, 2020, s. 18).

Konseptuaalisen mallin seuraavassa vaiheessa tietoaaltaaseen kerättyä raakadataa ladataan kohti datan prosessointivaihetta, joko jatkuvana syötteenä tai peräkkäisinä sarjoina riippuen tietueiden koosta, muodosta sekä tallennustiheydestä (Raj ja muut, 2020, s. 18). Ladattua dataa muokataan prosessointisilmukassa, jonka avulla prosessoitu data siirretään lopulta tietovarastoon (englanniksi Data Warehouse) lastausalueen (englanniksi Data Staging) kautta (Raj ja muut, 2020, s. 18). Prosessointisilmukan tehtävänä on siis muokata tietoaaltaan raakamuotoiset tiedostot data-analytiikan vaatimuksia tukemaan. Prosessointivaihe on eräänlainen yhdistelmä sisäkkäisiä silmukoita, joiden avulla dataa muokataan tarvittavin menetelmin, ja jonka lopputuloksena prosessoitu raakadata on muovattu yhtenäiseen muotoon data-analyysin tuottamista varten (Raj ja muut, 2020, s. 18). Valmis data siirretään lopuksi prosessoinnin jälkeen tietovarastoon, josta sitä voidaan hyödyntää eri datatuotteiden kuten raporttien tai koneoppimismallien kohdalla (Raj ja muut, 2020, s. 18). Tietovarasto on päätöksenteon tueksi luotu yhtenäinen datan varastointitila, jossa eri lähteistä kerätty data on muokattu dimensionaaliseen muotoon, joka muodoltaan tukee enemmän tosiaikaista tiedonjalostusta (Chaudhuri & Dayal, 1997, s. 65–66). Käsitelty data voidaan tarvittaessa tarkastaa ennen tietovarastoon siirtoa erillisellä lastausalueella, jossa sen oikeellisuudesta voidaan varmistua (Raj ja muut, 2020, s. 18).

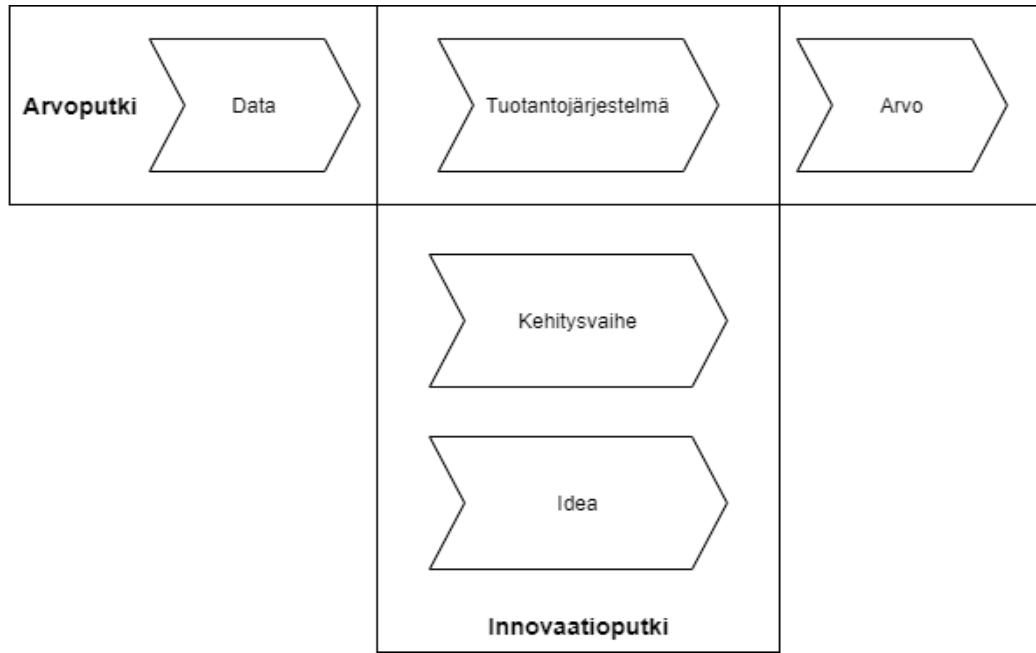
Rajin ja muiden (2020, s. 17–18) esittämään alkuperäiseen konseptuaaliseen malliin kuuluvat lisäksi tietovarastoinnin jälkeisinä silmukoina datan merkitseminen, esikäsittely sekä vastaanotto koneoppimismallissa, jossa dataa pystytään käyttämään esimerkiksi eri mallien harjoittamiseen. Koska nämä silmukat liittyvät enimmäkseen tämän tutkimuksen tarkastelun ulkopuolelle jääviin koneoppimismalleihin ja -ohjelmistoihin, ei niitä esitetä kuvion 3 yksinkertaistetussa konseptuaalisessa mallissa. On kuitenkin hyvä tiedostaa, jotta dataa on tarvittaessa mahdollista jatkojalostaa myös tietovarastointiin tallentamisen jälkeen.

2.3.2 Suoritus

Rakennusvaiheen jälkeen sen aikana kehitetty dataputki siirretään haluttuun ajoympäristöön varsinaista käyttöä varten (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Ajoympäristönä voidaan hyödyntää organisaation jo olemassa olevia resursseja sekä infrastruktuuria, joka voi auttaa hallitsemaan kustannuksia (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Toisaalta ympäristön skaalautuvuuden kannalta erilaiset pilvilaskentaan (englanniksi Cloud Computing) perustuvat pilvipalvelut (englanniksi Cloud Services) ovat varteenotettava vaihtoehto, joita hyödyntämällä tarvittavia resursseja, kuten laskentatehoa, saadaan nopeasti lisättyä tai laskettua vaihtuvan kysynnän mukaisesti (Balachandran ja muut, 2017, s. 1114–1116).

Suoritusvaiheen implementointi lainaa paljon käytänteitä DevOpsin maailmasta, sillä tässä vaiheessa tavoitteena on siirtää kehitystiimin rakentama dataputki kohti ajoympäristöä eli tuotannonjärjestelmiä (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Tähän kehityksen ja tuotannon välisen toimivuuden parantamiseen liittyy läheisesti DevOpsin käytänteistä tunnetut käsitteet jatkuva toimitus (englanniksi Continuous Delivery) sekä jatkuva integrointi (englanniksi Continuous Integration). Yksinkertaistettuna jatkuvan toimituksen ideana on tuottaa vähitellen uusia toiminnallisuuksia tuotannon järjestelmiin automaattisesti ja nopeasti lyhyillä kehityssykleillä (Capizzi ja muut, 2019, s. 1–4). Näitä automatisoituja toimituksia edeltää vastaavasti jatkuvan integroinnin vaihe, jossa eri kehittäjien tekemät muutokset, jotka usein yhdistetään ja kerätään yhtenäiseen

ohjelmistokoodin tallennus- ja versiointitilaan eli versionhallintaan (englanniksi Source Control), integroidaan yhdeksi yhtenäiseksi julkaisuksi (englanniksi Build) (Capizzi ja muut, 2019, s. 1–4). Tämä julkaisupaketti voidaan lopulta siirtää haluttuun kohdeympäristöön, jolloin kehityksen aikana toteutetut muutokset pystytään siirtämään laajempaan käyttöön hallitulla tavalla (Capizzi ja muut, 2019, s. 1–4). Vastaavanlainen automatisoitu prosessi sopii esimerkiksi Munappyn ja muiden (2020, s. 169) näkemyksen mukaan DataOpsin käytänteisiin, jossa samankaltaista ideaa pystytään hyödyntämään ajatteleamalla dataputken prosessin muodostuvan kahdesta erillisestä putkesta, joista ensimmäisen eli arvoa tuottavan putken (englanniksi Value Pipeline) tarkoituksena on dataputkiin sisältyvien toimintojen avulla tuottaa lisäarvoa lopputuloksena. Toisen osuuden eli innovaatiota tuottavan putken (englanniksi Innovation Pipeline) tehtävänä on vastavasti toimia prosessina, missä uudet analytiikan tarvitsemat toiminnot liitetään osaksi aikaisempaa arvoa tuottavaa putkea. Tämä DataOpsin prosessin jakaantuminen kahdeksi erilliseksi arvo- ja innovaatioputkeksi on nähtävissä myös Benghiatin ja muiden (2019, s. 30–32) näkemyksessä, jonka mukaan analyysien tuottamisen prosessi muodostuu kuvion 4 mukaisella tavalla:



Kuvio 4. Arvo- ja innovaatioputken prosessi uusien analyysien tuottamiseksi (mukaillen Benghiat ja muut, 2019, s. 32).

Kuvion 4 mukaisesti arvoa tuottavan putken tehtävänä analyysien tuottamisessa on kerätä dataa lähdejärjestelmistä, muokata se analyyseille sopivaan muotoon ja lopuksi siirtää prosessoitu data kohdejärjestelmään tuottamaan arvoa organisaatiolle, esimerkiksi analyysin tai raportin muodossa (Benghiat ja muut, 2019, s. 30–32; Munapy ja muut, 2020, s. 169, 173). Tämänkaltaisen arvoa jatkuvasti tuottava putki voidaan mieltää Rajin ja muiden (2020, s. 17–18) konseptuaalisen mallin mukaiseksi tuotantovalmiiksi dataputkeksi, joka on siirretty ajettavaksi tuotantojärjestelmän ajoympäristöön jatkuvan integroinnin sekä toimituksen vaiheiden avulla (Capizzi ja muut, 2019, s. 1–4). Prosessissa automatisoitu integrointi ja toimitus mahdollistetaan erillisellä innovaatioputkella, jossa esimerkiksi arvoputkesta johdettu uusi idea tai analytiikan tarvitsema toiminnallisuus kuljetetaan ketterän ohjelmistokehityksen periaatteiden mukaisesti kehitysvaiheen avulla osaksi aikaisempaa arvoa tuottavaa putkea (Benghiat ja muut, 2019, s. 30–32; Capizzi ja muut, 2019, s. 1–4). Erottelemalla analyysien tuottamisen prosessi kahdeksi erilliseksi toiminnoksi, pystytään dataputkien kehitys sekä niiden tuotantoon siirtäminen tuomaan lähemmäksi toisiaan systemaattisella tavalla, joka vastaavasti nopeuttaa uusien ideoiden sekä analyysien käyttöönottoa (Benghiat ja muut, 2019, s. 30–32).

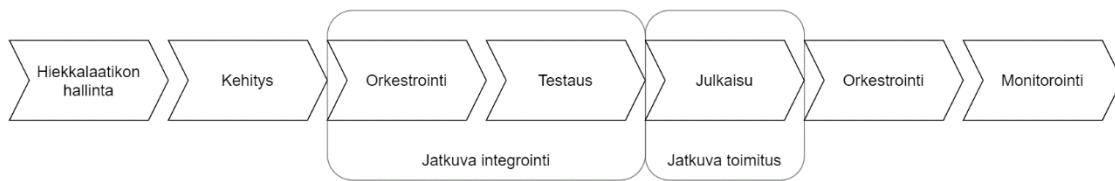
2.3.3 Käyttö

Implementoinnin viimeisessä vaiheessa dataputkien toimivuutta sekä niihin liittyvää ajo-ympäristöä tulisi monitoroida, jotta mahdollisiin ongelmiin pystytään reagoimaan (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Ongelmat voivat liittyä ympäristöön kuten laskentatehon riittävyyteen (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018) tai analyysien toimivuuteen ja luotettavuuteen (Benghiat ja muut, 2019, s. 10–11). Tämän käyttövaiheen jatkuvan monitoroinnin avulla saadaan kerättyä informaatiota prosessien ja menetelmien toimivuudesta, joka edesauttaa kehittämään organisaation käytänteitä DataOpsin tavoitteiden mukaisesti. Monitoroinnin tueksi Benghiat ja muut (2019, s. 20–22) esittävät käytettäväksi lean-ajattelusta peräisin olevaa tilastollista prosessikontrollia, jonka avulla data-analyysien tehokkuutta ja laatua voidaan parantaa asettamalla ennalta määriteltynä raja-arvoja, joiden välillä dataputkien tulisi toimia.

Käyttövaiheeseen kuuluu olennaisena osana myös testaus, joka DataOpsin kohdalla voidaan jakaa dataan sekä logiikkaan kohdistuviin testitapauksiin (Benghiat ja muut, 2019, s. 20–22, 25–26). Dataan kohdistuvassa testaamisessa testitapaukset kohdistetaan tarkastelemaan dataan liittyviä virheitä, kuten rivimääriä, kolumnien sisältämien arvojen oikeellisuutta tai johdonmukaisuutta (Benghiat ja muut, 2019, s. 22, 38–39). Vastaavasti logiikkaan perustuvassa testaamisessa tarkastelu painottuu validoimaan loogisia virheitä kuten analyysimallien lopputuloksien eroavaisuutta liiketoiminnan oletuksista (Benghiat ja muut, 2019, s. 22, 38–39). Tämä testien painopisteiden eroavaisuus näkyy myös arvo- ja innovaatioputkien välillä, joka Benghiatin ja muiden (2019, s. 38–39) tarkastelussa johtuu juuri DataOpsin mukanaan tuomasta datanäkökulmasta. Arvoa tuottavassa putkessa lähdejärjestelmistä siirretty lähdedata on muuttuva tekijä, jolloin testaaminen tulisi kohdistaa enemmän dataan liittyviin virhemahdollisuuksiin (Benghiat ja muut, 2019, s. 38). Vastaavasti kehityksen aikana rakenteilla olevan dataputken logiikka ja ohjelmistokoodi muuttuvat jatkuvasti, jolloin innovaatioputken kehitysvaiheessa testaaminen perustuu enemmän loogisen toiminnallisuuden varmistamiseen (Benghiat ja muut, 2019, s. 38).

2.3.4 Kehitys- ja julkaisuprosessi

Edellä esitetyt DataOpsin rakennus-, suoritus- sekä käyttövaihe muodostavat yhdessä yhtenäisen kokonaisuuden, jossa näitä vaiheita toistetaan jatkuvasti samassa järjestyksessä. Tämä DataOpsin kehitys- ja julkaisuprosessiksi nimetty kokonaisuus on esitetty alla olevassa kuviossa 5:



Kuvio 5. DataOps kehitys- ja julkaisuprosessi (mukaillen Benghiat ja muut, 2019, s. 37).

Kuvion 5 mukaisesti DataOpsin kehitys- ja julkaisuprosessi tarkoittaa implementoinnissa hyödynnettäviä päävaiheita jakamalla niitä useampiin vaiheisiin. Yksi näistä tarkennuksista on orkestrointi, jolla Benghiat ja muut (2019, s. 30–32) tarkoittavat tässä yhteydessä dataputken ominaisuutta, jossa sen sisältämiä eri toimintojen omaavia silmukoita järjestetään sekä aikataulutetaan käynnistymään vaiheittain. Tämänkaltaisen vaiheistus on nähtävissä erinomaisesti Rajin ja muiden (2020, s. 17–18) konseptuaalisen dataputken mallissa, jossa putken silmukat etenevät järjestyksessä yhdensuuntaisesti asyklisenä kaaviona. Silmukoiden orkestrointi tapahtuu niin kehitys- kuin julkaisuprosessissa, joka kuvastaa tarvetta integroida innovaatioita lisäävien dataputkien kehitys osaksi arvoa tuottavia tuotantojärjestelmissä ajettavia putkia (vrt. Benghiat ja muut, 2019, s. 30–32). Integraation toimivuus varmistetaan suoritusvaiheen mukaisella jatkuvan integroinnin ja toimituksen prosessilla, jossa kehitysympäristön muutokset on mahdollista testata ennen tuotantoon julkaisemista aikaisemmin mainittujen data- ja logiikkatestien avulla.

Merkittävimpänä lisäyksenä kehitys- ja julkaisuprosessissa esitetään ”hiekkalaatikon hallinta”. Benghiat ja muut (2019, s. 39–40) kuvaavat tarvetta erilliselle eristetylle kehitysympäristölle, jossa kehittäjät voivat rakentaa sekä testata uusia toiminnallisuuksia vaikuttamatta toisten kehittäjien aikaansaannoksiin. Data-analytiikan kohdalla

tämänkaltaisen täysin erillinen kehitysympäristö voi olla haastavaa toteuttaa, sillä etenkin käytettävän datan kopiointi voi tuottaa ongelmia tallennustilan, tietoturvan tai muiden dataan liittyvien rajoitteiden suhteen. Benghiat ja muut (2019, s. 40) näkevät edustavan testidatan luomisen ja siihen liitettävän automatisoidun kehitysympäristön pystyttämisen olevan edellytys kiihdyttämään analytiikan kehitystyötä.

3 Suunnittelutieteellinen tutkimusprosessi

Tässä tutkimuksessa DataOpsin periaatteita noudattavan dataputken suunnittelua ohjaa suunnittelutieteellinen tutkimusmenetelmä. Tietojärjestelmätieteessä suunnittelutieteellisen tutkimuksen tavoitteena on luoda sekä arvioida IT artefakteja, jotka pystyvät ratkaisemaan organisaation tunnistamat ongelmat (Hevner ja muut, 2004, s. 77). Hevner ja muut (2004, s. 77, 79–81) esittävät artefaktien olevan strukturoituja esiintymiä, joiden muoto voi vaihdella esimerkiksi ohjelmistosta muodolliseen logiikkaan tai täsmällisestä matematiikasta luonnollisen kielen kuvaukseen. Heidän mukaansa suunnittelutieteen tarkoituksena on lähestyä tietojärjestelmätieteen tutkimusta artefaktien rakentamisen sekä arvioinnin kautta, synnyttäen näin uusia innovaatioita tunnistettuihin mutta aikaisemmin ratkaisemattomiin haasteisiin. Suunnittelun tueksi Peffers ja muut (2008) loivat aikaisempaan tutkimukseen perustuen suunnittelutieteen DSRM-mallin, jonka tarkoituksena on toimia viitekehyksenä onnistuneelle suunnittelutieteelliselle tutkimukselle.

Tämä tutkimus hyödyntää suunnittelutieteellistä tutkimusta, luoden ensin aikaisempaan DataOpsin teoriaan pohjautuen listauksen dataputkelle tyypillisimmistä ja tärkeimmistä ominaisuuksista, kun putki suunnitellaan noudattamaan määritelmään liittyviä parhaita käytänteitä. Artefaktin luomisessa hyödynnetään suunnittelutieteellisen tutkimuksen DSRM-mallia, jonka demonstraativaiheessa artefaktin käyttöä simuloidaan kehittämällä pilviympäristössä ajettava dataputki listausta apuna käyttäen. Tutkimuksen lopputuloksena saadaan täten geneerinen listaus DataOpsin periaatteita mukailevan dataputken tyypillisimmistä ominaisuuksista, jotka putken suunnittelu- ja kehitysvaiheen aikana huomioimalla onnistutaan rakentamaan DataOpsin parhaita käytänteitä noudattava datan siirtämisen, muokkaamisen ja lataamisen prosessi. Ominaisuuslistaus muodostuu vahvasti jo aikaisemman teorian pohjalle, jolloin tutkimuksen tärkeimpänä kontribuutiona voidaan nähdä DataOpsin yleisen tietoisuuden lisääminen sekä demonstroitu dataputki, joka onnistuttiin luomaan DataOpsin parhaita käytänteitä mukailevaksi listausta hyödyntäen.

3.1 Tietopohja

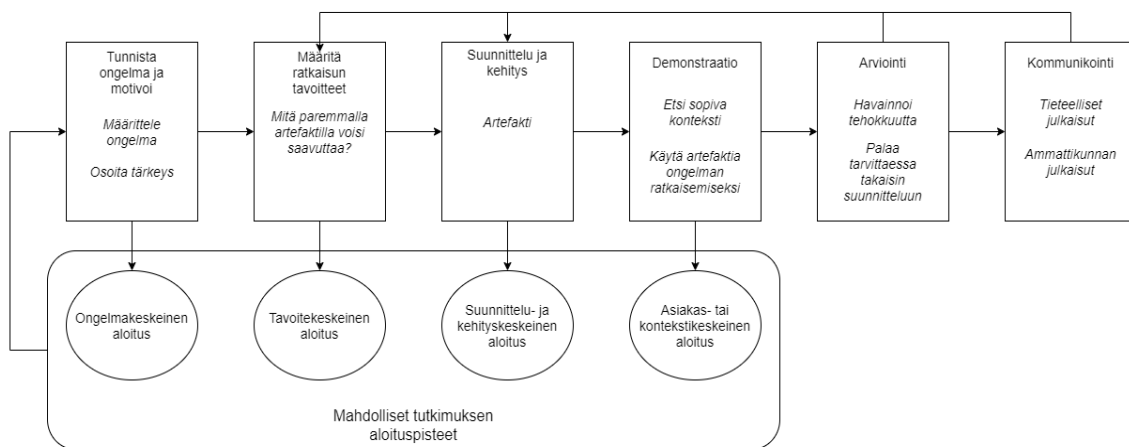
Suunnittelutieteellisessä tutkimuksessa aikaisempi tutkimustieto muodostaa pohjan, jonka avulla tutkimusta lähdetään rakentamaan (Hevner ja muut, 2004, s. 79–81). Aikaisemmin tutkittu tieto pitää sisällään perustan, joka koostuu teorioista, viitekehyksistä, välineistä, rakenteista, malleista, metodeista sekä muodostelmista. Itse tutkimusta ohjataan sekä arvioidaan erilaisilla menetelmillä, kuten data-analyysillä, formalisoinnilla, mitauksilla tai validoinnilla. Aikaisempi tutkimustieto yhdessä tutkimuksen arviointia ohjaavien menetelmien kanssa muodostavat suunnittelutieteelliselle tutkimukselle tietopohjan, johon uudella tutkimuksella tuotetaan lisäyksiä jatkokehitystä sekä -tutkimusta varten (Hevner ja muut, 2004, s. 79–81). Tutkimuksen tavoitteeksi Hevner ja muut (2004, s. 81) näkevät innovaatioiden luomisen tai tehokkaampien ratkaisujen kehittämisen organisaatioiden tunnistettuihin ongelmiin.

3.2 Ympäristö

Hevnerin ja muiden (2004, s. 79) mukaan aikaisemman tutkimustiedon perusteella, suunnittelutieteen ympäristö koostuu ihmisistä, organisaatioista sekä olemassa olevista teknologioista. Tavoitteet, tehtävät, ongelmat sekä mahdollisuudet muodostavat tarpeita liiketoiminnalle, jotka tulevat näkyviin organisaation ihmisten kautta. Hevner ja muut (2004, s. 79) jatkavat toteamalla näiden tarpeiden muokkaantuvan ihmisten roolien, kyvykkyyksien ja ominaisuuksien kautta organisaatiossa. Organisaation strategia, struktuuri, kulttuuri sekä jo olemassa olevat prosessit vastaavasti ohjaavat uusien liiketoimintatarpeiden arviointia. Nykyinen teknologinen infrastruktuuri taas asemoi uudet tarpeet jo toiminnassa olevien ratkaisujen, kuten applikaatioiden, arkkitehtuurien sekä kehityskyvykkyyksien suhteen (Hevner ja muut, 2004, s. 79). Ihmiset, organisaatio sekä teknologiat määrittävät yhdessä tarpeen tai ongelman, joka Hevnerin ja muiden (2004, s. 79) mukaan on havaittavissa. Tutkimuksen asiaankuuluvuus voidaan täten varmistaa tämän ajatuksen mukaisesti kohdistamalla tutkimuksen aktiviteetit osaksi liiketoiminnan tarpeita.

3.3 DSRM-prosessimalli

Suunnittelutieteellisen tutkimuksen tueksi luotu suunnittelutieteen tutkimusmalli (englanniksi Design Science Research Model, DSRM) on Peffersonin ja muiden (2008) kehittämä viitekehys ohjaamaan tutkimusprosessia. Mallin tarkoituksena on tarjota yleisesti hyväksytty viitekehys suunnittelutieteelliselle tutkimukselle (Pefferson ja muut, 2008, s.48). Aikaisemman tiedon pohjalta suunniteltu DSRM-malli pitää sisällään yhteensä kuusi erilaista aktiviteettia alla olevan kuvion 6 mukaisesti.



Kuvio 6. Suunnittelutieteen tutkimusmalli eli DSRM-malli (mukaillen Peffers ja muut, 2008, s. 54).

3.3.1 Ongelman määrittäminen ja motivointi

Pefferson ja muut (2008, s. 52–55) näkevät ongelman tarkan määrittämisen osana artefaktin kehitysprosessia, sillä ongelman määrittäminen hyödynnetään sen ratkaisuun tarkoitettujen artefaktin kehityksessä. Nostamalla esille ratkaisun tuomaa lisäarvoa, tutkijoita sekä tutkimusyleisöä voidaan motivoida etsimään ratkaisua sekä hyväksymään saavutetut tulokset. Lisäksi tämä auttaa muuta yhteisöä ymmärtämään tutkijan omaa näkemystä sekä päättelyä itse ongelmasta. Pefferson ja muut (2008, s. 55) jatkavat toteamalla tämän vaiheen resurssien sisältävän tietämystä itse ongelman tilasta sekä sen ratkaisemisen tärkeydestä.

3.3.2 Tavoitteiden asettaminen

Ratkaisun tavoitteiden asettaminen tulisi muodostaa ongelman määrittelyn pohjalta sekä tietämykseen siitä mikä on mahdollista (Pefferson ja muut, 2008, s. 55). Asetetut tavoitteet voivat olla määrällisiä, esimerkiksi parannuksia aikaisempiin ratkaisuihin tai laadullisia, kuten kuinka uuden ratkaisun oletetaan tukevan aikaisemmin ratkaisemattomia ongelmia. Pefferson ja muut (2008, s. 55) näkevät näiden tavoitteiden tarvitsevan resursseikseen tietoa ongelman nykytilasta sekä siihen jo hyödynnettävistä ratkaisuksista ja niiden tehokkuudesta.

3.3.3 Artefaktin suunnittelu

DSRM-mallin kolmantena vaiheena on uuden artefaktin suunnittelu ja kehitys (Pefferson ja muut, 2008, s. 55). Suunniteltu artefakti voi olla mikä tahansa suunniteltu objekti, jonka sisään on liitetty tutkimuksen tuottamaa lisäarvoa. Tällaisina objekteina voidaan pitää esimerkiksi ohjelmistoja, muodollista logiikkaa, täsmällistä matematiikkaa tai luonnollisen kielen kuvausta, kuten Hevner ja muut (2004, s.77) luettelevat strukturoituiksi esiintymiksi. Pefferson ja muut (2008, s. 55) näkevät tämän vaiheen sisältävän kuvausta suunnitellun artefaktin toiminnallisuudesta sekä sen arkkitehtuurista. Tähän näkemykseen sisältyy myös itse artefaktin kehitys, jonka resurssiksi he katsovat tarvitsevan tietämystä mahdollisesta ratkaisusta.

3.3.4 Demonstrointi

Demonstrointivaiheessa suunniteltua artefaktia tulisi käyttää ratkaisemaan itse ongelma (Pefferson ja muut, 2008, s. 55). Pefferson ja muut (2008, s. 55) esittävät mahdollisiksi demonstraatiokeinoiksi kokeilut, simulaatiot, tapaustutkimukset tai esimerkiksi todistukset. Demonstrointi vaatii tietämystä suunnitellun artefaktin käytöstä ongelmanratkaisemiseksi (Pefferson ja muut, 2008, s. 55).

3.3.5 Arviointi

Viidentenä vaiheena Pefferson ja muut (2008, s. 56) nostavat esille kehitetyn artefaktin arvioinnin. Arvioinnissa tulisi kiinnittää huomiota siihen, kuinka hyvin suunniteltu artefakti tukee ongelman ratkaisua. Arviointi voidaan toteuttaa monella eri tavalla, riippuen artefaktin luonteesta, jonka Pefferson ja muut (2008, s. 56) näkevät vaikuttavan arviointityökalun valintaan. Esimerkkeinä he mainitsevat kehitetyn artefaktin toiminnallisuuksien vertailemisen aikaisemmin asetettuihin tavoitteisiin, tyytyväisyyskyselyjen tulokset, käyttäjien palautteen, simuloinnit sekä määrälliset mittaukset, kuten budjetoinnin tai tuotettujen tuotteiden määrän. Konseptuaalisella tasolla Pefferson ja muut (2008, s. 56) esittävät arvioinnin voivan koostuvan mistä tahansa sopivasta empiirisestä todisteesta tai loogisesta todistuksesta. Tietojärjestelmien kohdalla järjestelmän tehokkuuteen kohdistuvat mittaukset, kuten vasteajat tai saatavuudet ovat heidän mukaansa myös toimivia arvioinnin kohteita.

Hevner ja muut (2004, s. 82–90) esittävät suunnittelutieteellisen tutkimuksen arvioinnin tueksi yhteensä seitsemän ohjetta, jotka auttavat tutkijoita tehokkaaseen tutkimukseen. Nämä ohjeet ovat tiivistettynä alla ja niitä sovelletaan tämän tutkimuksen arviointiin soveltuvin osin tutkielman viimeisessä kappaleessa:

1. Suunnittelutieteellisen tutkimuksen tulee tuottaa käyttökelpoinen artefakti rakenteellisessa muodossa, mallissa, metodissa tai ilmentymänä.
2. Suunnittelutieteellisen tutkimuksen tavoitteena on kehittää teknologiaan pohjautuvia ratkaisuja tärkeisiin ja asiaankuuluviin liiketoiminnan ongelmiin.
3. Artefaktin hyödyllisyys, laatu ja tehokkuus on osoitettava hyvin toteutettujen arviointimenetelmien avulla.
4. Tehokkaan suunnittelutieteellisen tutkimuksen on tuotettava selkeää ja todennettavissa olevaa kontribuutiota suunniteltuun artefaktiin, suunnittelun perusteisiin ja/tai metodologieihin.
5. Suunnittelutieteellinen tutkimus perustuu täsmällisiin metodeihin artefaktin rakennus- ja arviointivaiheissa.

6. Tehokkaan artefaktin etsiminen edellyttää käytettävissä olevien keinojen hyödyntämistä tyydyttävän ratkaisun löytämiseksi, huomioimalla kuitenkin ympäristön asettamat lainalaisuudet.
7. Suunnittelutieteellinen tutkimus on esitettävä tehokkaasti sekä teknologian asiantuntijoille kuin myös yritysjohdolle.

Arvioinnin lopuksi tutkijat voivat arvioinnin tulosten perusteella pyrkiä parantamaan kehitettyä artefaktia (Pefferson ja muut, 2008, s. 56). DSRM-mallin mukaisesti artefaktin suunnitteleminen on iteratiivinen prosessi, jossa arvioinnista saadun palautteen perusteella tutkijat voivat palata mallin aikaisempiin vaiheisiin parantaakseen artefaktin tehokkuutta ratkaistavaan ongelmaan nähden (Pefferson ja muut, 2008, s. 54, 56). Toisena vaihtoehtona on edetä viimeiseen vaiheeseen ja jättää mahdolliset jatkokehitykset tuleville projekteille ja tutkimuksille.

3.3.6 Viestintä

Viimeisenä DSRM-mallin vaiheena on suunnittelutieteellisen tutkimuksen kommunikointi laajemmalle yleisölle, kuten tutkimusyhteisölle ja ammatinharjoittajille (Pefferson ja muut, 2008, s. 56). Kommunikoinnissa tulisi esitellä tutkimuksen aiheena esiintyvä ongelma ja sen tärkeys, suunniteltu artefakti sekä artefaktin tuoma hyöty, uutuusarvo ja tehokkuus olennaisille sidosryhmille (Pefferson ja muut 2008, s. 56). DSRM-malli itsessään voidaan nähdä juuri tämänkaltaisena kommunikointivälineenä suunnittelutieteelliseen tutkimukseen liittyvissä tieteellisissä julkaisuissa, kuten Pefferson ja muut (2008, s. 56) toteavat.

4 Dataputken kehitysprosessi

Tässä kappaleessa kuvataan dataputken kehitysprosessi, jossa ensin tutkielman teoreettisen viitekehyksen pohjalta luodaan listaus tyypillisimmistä ominaisuuspiirteistä, joita DataOpsin parhaita käytänteitä noudattava dataputki sisältää. Muodostetun listan avulla suunnitellaan varsinainen dataputki, jonka käyttöä sekä kehitystä demonstroidaan Azuren pilviympäristössä. Lopputuloksena saadaan siten ominaisuuslistaus, jonka avulla DataOpsin periaatteita mukailevan datan siirtämisen, muokkaamisen ja lataamisen prosessi eli dataputki on mahdollista kehittää.

4.1 Tietopohja

Artefaktin suunnittelu perustuu vahvasti jo aikaisempaan olemassa olevaan tutkimustietoon eli DataOpsin määritelmään ja sen käytänteisiin, joita on avattu tarkemmin tämän tutkimuksen teoriakatsauksessa. Lisäksi tietopohja painottaa käsitteeseen liittyviä teknologisia aihealueita eli tutkielman tarkastelu on erittäin teknologiakeskeinen (vrt. Ereth, 2018, s. 5).

4.2 Ympäristö

Suunnitellun artefaktin teknologisenä ympäristönä toimii Microsoftin tarjoama pilvipalvelu nimeltään Azure sekä sen tarjoamat resurssit (ks. Microsoft, 2021a). Palvelu valittiin tutkimuksen ympäristöksi tutkijan omien kokemusten pohjalta, joita avataan tarkemmin seuraavassa aktiviteetissa. Pilvipalvelun etuina voidaan nähdä skaalautuvuus kysynnän mukaan (Balachandran ja muut, 2017, s. 1114–1116) sekä tarvittavien resurssien nopea saatavuus, jotka edesauttavat monia DataOpsin tavoitteita kuten nopeasti tuotettavaa data-analyysin valmistamista tai automatisoinnin lisäämistä dataputken kehitys- ja suorituvaiheissa.

Tutkija tuotti suunnitellun artefaktin toimiessaan datan hallintaan liittyvissä työtehtävissä mutta kuitenkin omista tehtävistään täysin irrallisena kokonaisuutena työn ulkopuolella. Tarkoituksena oli toteuttaa täysin itsenäinen sekä mahdollisimman geneerinen artefakti, jota pystyisi tarvittaessa hyödyntämään eri liiketoimintasektoreilla. Tutkielman ympäristönä organisaation ja ihmisten suhteen voidaan pitää datan hallintaa, tarkemmin data-analytiikan tuottamista liiketoimintasektorista riippumatta. Ympäristön lähtökohta on siten erittäin teknologiapainotteinen.

4.3 Ongelman määrittäminen ja motivointi

Tutkimuksen isoimpana motivaationa toimi tutkijan omakohtaiset kokemukset data-analytiikkaan kohdistuvista ongelmista. Analyyseissä hyödynnettävien dataputkien kehitystyö on usein todella manuaalista sekä tietyllä tavalla kertaluonteista työtä, jossa tarvittavien muutoksien tekeminen jo tuotannossa pyöriviin ratkaisuihin koetaan riskialttiiksi toiminnaksi. Useasti haasteet liittyvät testaamisen puutteisiin, joka hidastaa uusien tarpeiden kehitystä sekä päivitetyn data-analyysin tuottamista. Testaamisen ongelmat ovat esillä myös dataan liittyvässä kirjallisuudessa, josta hyvänä esimerkkinä on tutkimuksen teoriaosuudessa esitetty Benghiatin ja muiden (2019, s. 8–12) listaus data-analyysin kohtaamista yleisistä haasteista.

Toisena isona motivaationa artefaktin suunnittelulle toimi nykyajan liiketoiminnassa tarvittava nopeus ja ketteruus. Kuten Benghiat ja muut (2019, s. 8–12) erinomaisesti esittävät, muutos liiketoiminnassa on jatkuva tila, joka osaltaan vaatii organisaatioita reagoimaan nopeasti havaittuihin muutostarpeisiin pysyäkseen kilpailukykyisinä. Nopea dataan liittyvä päätöksenteko vaatii tuekseen nopeasti toimitettavaa data-analytiikkaa, jolloin organisaation on kyettävä tekemään tarvittavia muutoksia analyysihin nopeasti sekä luotettavasti. Yksinkertainen listaus, jota seuraamalla pystyttäisiin toistuvasti tuottamaan toimivia dataputkia, olisi täten hyödyllinen työkalu esimerkiksi kehittäjille muistuttamaan oikeanlaisesta suunnasta jokapäiväisessä kehitystyössä. Parhaassa tapauksessa tutkija itse pystyisi hyödyntämään listausta sekä sen avulla suunnitellun

dataputken runkoa tavalla, joka mahdollistaisi kehityksen aloittamisen aina samalla automatisoidulla prosessilla liiketoimintasektorista, asiakkaasta tai sidosryhmästä riippumatta.

Kolmas motivaatiotekijä artefaktin kehitykselle oli DataOpsin tavoite prosessien automatisoitavuudesta. Usein dataputken suunnittelussa toistuvat samat siirtämisen, muokkaamisen ja lataamisen pääpiirteet, jolloin idea tämän prosessin automatisoinnista herätti tutkijan mielenkiinnon. Toistettava dataputki, joka pystyttäisiin automaattisesti siirtämään kehityksestä tuotantoympäristöön, vähentäisi merkittävästi manuaalisen työn sekä siihen liittyvien riskien osuutta. Lisäksi jo aikaisemmin mainitun testaamisen integroiminen osaksi tätä automatisoitua siirtoa varmistaisi jo aikaisemmin muodostettujen putkien toimivuuden. Tämä osaltaan vapauttaisi aikaa uusien vaatimusten sekä raporttien kehitystyölle.

DataOpsin lupaukset parhaista käytänteistä, joiden avulla pystytään ratkaisemaan data-analyysin kohtaamat erilaiset haasteet, tarjoaa siksi varteenotettavan lähtökohdan data-analytiikassa hyödynnettävien dataputkien suunnittelu- ja kehitystyölle. Tämän tutkimuksen tarkoituksena on osaltaan esittää DataOpsin tarjoamat mahdollisuudet, sekä kuinka siihen liittyviä parhaita käytänteitä onnistutaan jalkauttamaan yksinkertaisella tavalla kehittäjien jokapäiväiseen kehitystyöhön, kun tarkastellaan määritelmään liittyviä teknologisia näkökulmia.

4.4 Tavoitteiden asettaminen

Artefaktin tavoitteena on toimia ylätasen ominaisuuslistauksena, jota seuraamalla data-analytiikassa hyödynnettävän datan lataamiseen, muokkaamiseen ja siirtämiseen tarkoitettua dataputken suunnittelu- ja kehitystyötä voidaan ohjata kohti DataOpsin mukaisia parhaita käytänteitä. DataOpsin periaatteita mukaileva dataputki voisi siten tarjota ratkaisuja teoriakatsauksessa mainittuihin data-analytiikan kohtaamiin haasteisiin. Listaus pidetään mahdollisimman geneerisenä, jolloin se ei ota kantaa käytettäviin

teknologioihin vaan sen tarkoituksena on toimia enemmän kehittäjien apuna dataputkien kehitystyön yhteydessä.

Konkreettisena tutkimuksen tavoitteena on tuottaa toimiva esimerkki DataOpsin käytänteitä noudattavasta dataputkesta, joka voidaan kuvailla tutkimuksessa luodun artefaktin eli ominaisuuslistauksen avulla.

4.5 Artefaktin suunnittelu ja kehitys

Artefaktin suunnittelu aloitettiin tutustumalla tutkimuksen teoriakatsauksessa esitettyihin DataOpsin erilaisiin määritelmiin sekä manifeston periaatteisiin. Tavoitteena oli saavuttaa aikaisempaan tutkimustietoon pohjautuen ylätason yhteenveto tärkeimmistä ominaisuuksista, joita seuraamalla dataputken suunnittelu- ja kehitystyötä voitaisiin lähteä toteuttamaan. Teoriakatsauksen avulla esille nousi neljä selkeää ominaisuutta, joista jokainen oli löydettävissä DataOpsin määritelmistä sekä siihen liittyvästä muusta teoriasta. Ominaisuudet olivat seuraavat:

Toistettavuus.

Ominaisuutena toistettavuus on vahvasti esillä dataputken konseptuaalisessa mallissa (Raj ja muut, 2020, s. 17–18). Mallin mukaisesti dataputki tulisi suunnitella ja kehittää siten, jotta alkuperäinen raakadata pystyttäisiin tarvittaessa viemään uudelleen dataputken lävitse. Dataputken lopputuloksen tulisi olla identtinen jokaisella ajokerralla, vaikka putki jouduttaisiin käynnistämään uudelleen myöhemmin, esimerkiksi ajovirheen seurauksena. Sama toistettavuus voidaan laajentaa koskemaan myös ajoympäristöä, jolloin tarkoituksena on dataputken suorittamisen mahdollistaminen toisessa samankaltaisessa ympäristössä. Tämänkaltainen toistettavuus on lisäksi yksi mahdollistava tekijä automatisoinnin käyttöönotolle. Kun ympäristöt onnistutaan rakentamaan ilman manuaalisia vaiheita tai muokkauksia, pystytään niiden luominen toistamaan automatisoidulla

prosessilla, joka osaltaan tukee DataOpsin tavoitetta automatisaation lisäämisestä (Ereth, 2018, s. 5; Munappy, 2020, s. 169).

Toistettavuus mainitaan myös DataOps manifeston sisältämissä periaatteissa yksitoista sekä seitsemäntoista (DataOps Manifesto, 2021). Näistä ensimmäinen korostaa versiointia esimerkiksi ohjelmistokoodin ja siihen liittyvien työkaluketjujen suhteen, jolla mahdollistetaan toistettavat tulokset. Jälkimmäinen taas ohjeistaa välttämään aikaisemmin tehdyn työn toistamista analytiikan valmistuksen tehostamiseksi uudelleenkäyttämällä kehitettyjä ratkaisuja.

Automatisoitavuus.

Automatisoinnin korostaminen näyttäytyy vahvana tekijänä monessa DataOpsia koskevassa ajatuksessa. Palmer (2015) näkee automatisoinnin olevan osa datanhallinnan menetelmää, Ereth (2018, s. 5) kuvailee määritelmässään DataOpsin hyödyntävän ketterän ohjelmistokehityksen tarjoamia automatisoinnin menetelmiä, kun vastaavasti Sahoo ja muut (2019, s. 1–2) näkevät DevOpsin pohjalta perityt automatisoidut prosessit tärkeänä datasta saatavana lisäarvon tuottajana. DevOps-keskeinen automatisoitu näkökulma on havaittavissa myös Capizzin ja muiden lopputulemassa (2019, s. 1–5) sekä Benghiatin ja muiden (2019, s. 22) näkemyksessä, jossa DevOps on yksi kolmesta DataOpsin osatekiijästä. Myös Munappy ja muut (2020, s. 169) liittävät automatisoinnin roolin osaksi heidän määritelmäänsä. Dataputki sekä siihen liittyvä ajoympäristö tulisi täten suunnitella tukemaan mahdollisimman pitkälle prosessien automatisointia, jolloin DevOpsin maailmasta tutut käytänteet, kuten jatkuva integrointi sekä jatkuva toimitus, olisi hyvä ottaa huomioon jo dataputken suunnitteluvaiheen aikana (ks. Capizzi ja muut, 2019, s. 1–4).

Testattavuus.

Dataputkiin kohdistuva testaaminen korostuu etenkin Benghiatin ja muiden (2019, s. 20–22, 25–26) kokemuksissa, jossa oikeanlaisella testaamisella pystytään tuottamaan

varmuutta kehitystyöhön. Tämä osaltaan edesauttaa muutoksien onnistunutta siirtämistä kohti tuotantoympäristöä, jonka he näkevät yhtenä osana uuden arvon luontia analyysin tuottamisen arvo- ja innovaatioputken prosessissa (Benghiat ja muut, 2019, s. 32). Toimivaa testausta voidaan pitää myös edellytyksenä korkealaatuisten tuloksien aikaansaamiseksi, joita esimerkiksi Munappy ja muut (2020, s. 169) kuvailevat osana DataOpsin tavoitteista.

Seurattavuus.

Jotta mahdollisiin ongelmiin pystytään reagoimaan, organisaatioilla tulisi olla mahdollisuus seurata ja monitoroida dataputkien suoritusta. Capizzi ja muut (2019, s. 4–5) sekä Patterson (2018) jakavat tämän näkemyksen kiinnittämällä huomiota monitorointimahdollisuuksiin niin ajoympäristöjen kuin niiden sisällä ajettavien dataputkien suhteen. Samankaltaista seurannan tärkeyttä korostavat Benghiat ja muut (2019, s. 20–22) tuodessaan tilastollisen prosessikontrollin menetelmät tukemaan DataOpsin periaatteita. Seurattavuus mainitaan myös DataOps manifeston kuudennentoista periaatteen kohdalla, jossa suorituskäy-, turvallisuus- ja laatutoimenpiteitä neuvotaan monitoroimaan jatkuvasti (DataOps Manifesto, 2021). Näin onnistutaan tuottamaan toiminnallisia tilastoja sekä havaitsemaan odottamattomat vaihtelut prosesseissa.

Näiden DataOpsin teoriassa havaittujen toistuvien ominaisuuksien eli toistettavuuden, automatisoitavuuden, testattavuuden ja seurattavuuden pohjalta luotiin yhteenvetona tutkielman artefakti. Artefakti on esitetty alla olevassa taulukossa 1:

Taulukko 1. Tyypillisimmät ominaisuudet DataOpsin periaatteita noudattavalle dataputkelle ja sen ajoympäristölle.

Ominaisuus	Dataputki	Ajoympäristö
Toistettavuus	Onko dataputki toistettavissa toisessa ympäristössä? Onko dataputki siirrettävissä toiseen ympäristöön? Onko dataputken ajo toistettavissa?	Onko ajoympäristö toistettavissa toisessa ympäristössä? Onko ajoympäristö siirrettävissä toiseen ympäristöön?
Automatisoitavuus	Onko dataputken ajo automatisoitavissa? Onko dataputkeen liittyvien muutostöiden julkaiseminen automatisoitavissa?	Onko ajoympäristön luominen automatisoitavissa?
Testattavuus	Onko dataputki testattavissa?	Onko ajoympäristö testattavissa?
Seurattavuus	Onko dataputken ajo seurattavissa?	Onko ajoympäristön tila seurattavissa?

Taulukossa 1 esitetään siis yksinkertaistettu ylätasoinen listaus DataOpsin dataputkeen sekä sen ajoympäristöön liittyvistä tärkeimmistä sekä tyypillisimmistä ominaisuuksista. Lisäksi taulukon ominaisuuksiin on liitetty muutamia tarkentavia kysymyksiä, joiden tarkoituksena on auttaa kehittäjiä ohjaamaan putkien sekä niiden ajoympäristöihin liittyvää suunnittelua- sekä kehitystyötä. Huomionarvoista on, ettei listaa tulisi lukea ainoana oikeana eikä kaikenkattavana välineenä DataOpsin periaatteita noudattavan datan siirtämiseen, muokkaamiseen ja lataamiseen tarkoitetun dataputken suunnittelutyössä. Artefaktia tulisi lähestyä enemmän päivittäisenä yleistä suuntaa antavana ohjeena kehittäjille, joka kuitenkin mahdollistaa DataOpsin parhaita käytänteitä noudattavan dataputken sekä siihen kohdistuvan ajoympäristön luomisen ja kehittämisen teknologiariippumattomasti. Tästä esimerkkinä on tutkimuksen seuraavassa aktiviteetissa esitetty demonstraatio artefaktin käytöstä dataputken suunnittelu- sekä kehitysvaiheessa.

4.6 Demonstrointi

Artefaktin avulla suunnitellun dataputken kehitystyö aloitettiin tutustumalla tutkielman teknologisenä ympäristönä toimivan Azure pilvipalvelun tarjoamiin resursseihin, jotka mahdollistaisivat datan siirtämisen lähdejärjestelmästä yhtenäisen tallennuspaikan kautta kohti data-analytiikan tarvitsemaa tallennustilaa Rajin ja muiden (2020, s. 17–18) konseptuaalisen mallin periaatteita noudattavalla tavalla. Azuren tarjoamat resurssit datan käsittelyyn ovat valtavat sekä jatkuvasti kehittyvät, mutta niitä kartoitettaessa päädyttiin dataputken suunnittelu kohdistaa seuraavien palveluiden varaan:

- a) Azure Data Factory
- b) Azure DevOps
- c) Azure Blob Storage
- d) Azure Database for PostgreSQL server
- e) Flyway

Azure Data Factory on Microsoftin tarjoama pilvipohjainen ETL-prosessien ja niiden hallinnoimiseen perustuva palvelu, jonka avulla pystytään kehittämään datan siirtämiseen, muokkaamiseen ja lataamiseen tarvittavia aktiviteetteja (Microsoft, 2021b). Työkalun avulla pystytään lisäksi luomaan yhteyksiä tarvittaviin resursseihin kuten tutkielmassa hyödynnettävään tietokantaan. Data Factory on myös mahdollista integroida kiinni versionhallintaan, joka mahdollistaa jatkuvan integroinnin sekä toimituksen prosessien liittämisen osaksi dataputkea DataOpsin periaatteiden mukaisesti. Versionhallinta sekä jatkuvan integrointi ja toimitus mahdollistetaan Azuren tarjoamalla Azure DevOps palvelulla, johon sisältyy versionhallinta Azure Repos-nimisenä resurssina sekä jatkuvan integroinnin ja toimituksen tuottamiseen tarkoitettu Azure Pipelines palvelu (Microsoft, 2021c). Palvelu tarjoaa lisäksi mahdollisuudet ohjelmistokoodin testaamiseen joko erillisellä Azure Test Plans resurssilla tai vaihtoehtoisesti hyödyntämällä sisäänrakennettua testausaktiviteettia jo aiemmin mainitun Azure Pipelines tuotteen sisällä.

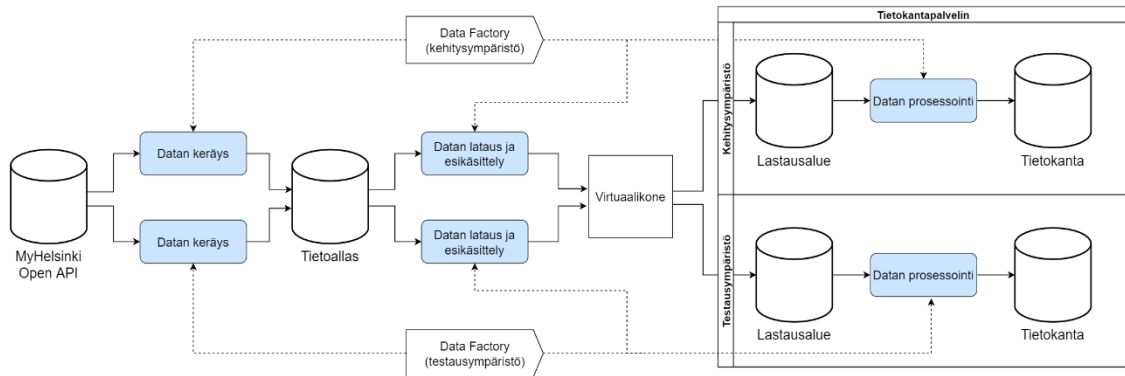
Suunnitellun dataputken datan käsittelyyn liittyvät aktiviteetit muodostuvat suurimalta osin näiden kahden palvelun eli Azure Data Factoryn sekä Azure DevOpsin

ympäri, joiden nähtiin toteuttavan tutkielman artefaktin esittämät vaatimukset Data-Opsin periaatteita noudattavalle dataputkelle. Ominaisuuksista ainakin toistettavuus, automatisoitavuus sekä testattavuus koettiin olevan toteutettavissa kysymysten mukaisilla tavoilla näiden kahden tuotteen tarjoamien kyvykkyyksien avulla. Listauksen ominaisuuksista seurattavuus koettiin myös mahdolliseksi toteuttaa Data Factoryn sisäänrakennetun monitorointitoiminnon avulla.

Seuraavana vaiheena dataputken suunnittelussa tehtiin valinta käytettävistä datan tallennusmenetelmistä. Lähdejärjestelmästä siirretyn alkuperäisessä muodossaan olevan tietueen eli raakadatan tallennustilaksi valikoitui Azure Blob Storage palvelu, joka tarjoaa pilvipohjaista tallennustilaa datan tallentamiseen miltei missä muodossa tahansa (Microsoft, 2021d). Tutkielman datalähde tuotti json-tiedostomuotoista eli puolistrukturoitua dataa, jonka tallentaminen sellaisenaan oli mahdollista palvelun avulla. Seuraavan vaiheen eli prosessoinnin jälkeiseksi tallennustilaksi päädyttiin hyödyntämään PostgreSQL tietokantaa, jonka Microsoft tarjoaa käytettäväksi Azure Database for PostgreSQL server -nimisenä tuotteena Azuren pilviympäristössä (Microsoft, 2021e). Nykyisin suurimmat pilvipalvelut tarjoavat perinteisiin relaatiokantoihin nähden data-analytiikkaan räätälöityjä tuotteita kuten pilvipohjaisia tietovarastoja. Näiden avulla suurien datamassojen käsittely voi olla huomattavasti tehokkaampaa verrattuna vakiintuneempiin ja enemmän tosiaikaista transaktiokäsittelyä tukeviin relaatiokantapohjaisiin tietokantatuotteisiin. Tässä tutkielmassa päädyttiin kuitenkin hyödyntämään prosessoidun datan tallennustilana tietokantaa, joka tukee strukturoidun datan lisäksi myös puolistrukturoitua dataa. PostgreSQL (PostgreSQL, 2021) on avoimeen koodiin pohjautuva tietokantajärjestelmä, joka tarvittaessa skaalautuu myös teratavujen kokoisiin datamääriin. Tämän tutkielman datamäärällä sen todettiin olevan täysin riittävä kapasiteetti, etenkin tutkimuksen painottaessa dataputken suunnittelua eikä datan mallintamista tai loppukäyttöä data-analytiikan raporteissa, joiden kohdalla riittämätön laskentakapasiteetti voi aiheuttaa isoja haasteita. On kuitenkin hyvä huomata, jotta esimerkiksi pilvipohjaiset tietovarastot ovat todennäköisempi valinta datamäärien kasvaessa huomattavasti yli teratavujen kokoisiin datamääriin organisaatioissa.

Tietokannan valinta aiheutti kuitenkin ensimmäisen isomman haasteen dataputken suunnittelussa, sillä artefaktin listausta läpikäytäessä huomattiin, jotta tietokannan liittäminen osaksi jatkuvaa integrointia sekä toimitusta ei suoraan toiminut osana Azure Pipelines palvelua. PostgreSQL tietokannan julkaisemiseen liittyvät toiminnot eivät tutkimuksen tekemisen aikana olleet osa Pipelines resurssin tarjoamia tietokantojen julkaisumenetelmiä. Jotta artefaktin listaamista ominaisuuksista toistettavuus sekä automatisoitavuus pystyttäisiin toteuttamaan, päädyttiin lopulta hyödyntämään Flyway-nimistä ohjelmistoa osana jatkuvan integroinnin ja toimituksen prosessia. Flyway on yksinkertaisuutta painottava ja avoin tietokantojen migraatioon tarkoitettu ohjelmisto, jonka käyttäminen on mahdollista Azure Pipelines resurssin sisällä, esimerkiksi konttitekniologiaa hyödyntävän Docker palvelun avulla (Flyway, 2021). Yksinkertaistettuna konttitekniologialla tarkoitetaan ohjelmistokoodin ja siihen liittyvien kaikkien riippuvuuksien pakkaamista yhtenäiseksi standardoiduksi yksiköksi eli kontiksi tavalla, jonka ansiosta sitä pystytään ajamaan nopeasti ja luotettavasti eri ympäristöissä (Docker, 2021). Flywayta päädyttiin käyttämään tietokannan julkaisuun liittyvissä aktiviteeteissa kutsumalla työkalua Dockeriin sisältyvän toiminnon avulla. Konttitekniologian tarkempi kuvaus on rajattu tämän tutkimuksen ulkopuolelle, sillä Flywayta käytetään tässä tutkimuksessa vain sen toiminnallisuuden vuoksi, joka ei sinällään liity konttitekniologian hyödyntämiseen.

Näiden Azuresta löydettyjen palveluiden ja resurssien avulla suunniteltiin lopulta malli DataOpsin periaatteita hyödyntävälle dataputkelle, jossa valitusta lähdejärjestelmästä voitaisiin siirtää dataa raakadatan tallennustilan eli tietotalteen kautta kohti prosessoidun datan tallennusalueeksi toimivaa tietokantaa. Alla oleva kuvio 7 esittää suunnitellun dataputken mallin:



Kuvio 7. Suunniteltu dataputki ajettavaksi kehitys- ja testausympäristössä Azuren pilvipalvelussa.

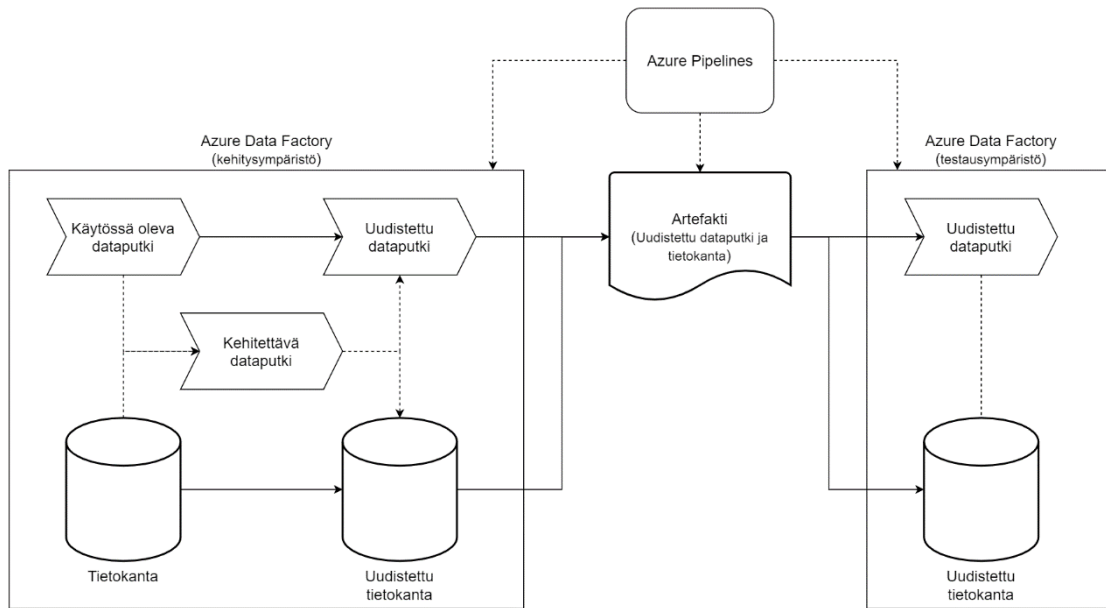
Kuvion 7 mukaisesti suunniteltu dataputki hyödyntää Rajin ja muiden (2020, s. 17–18) ajatuksia toimivan dataputken suunnittelussa. Lähdejärjestelmänä tässä tutkimuksessa hyödynnettiin MyHelsinki-sivuston tarjoamaa avointa rajapintaa, jonka kautta noudettiin tapahtumatietoihin liittyvää json-pohjaista eli puolistrukturoitua dataa (MyHelsinki, 2021). Nämä tapahtumatiedot siirrettiin raakamuotoisena tietoaltaaseen eli Azure Blob Storagen tarjoamaan tallennustilaan. Tietoaltaasta raakadata ladattiin samaan dataputkeen sisältyvän toisen aktiviteetin avulla kohti tietokannan erillistä lastausaluetta, jossa siihen kohdistettiin alustava esikäsittely jakamalla tiedoston sisältämä data kahteen osaan: varsinaisiksi tapahtumatiedoiksi sekä näiden tapahtumatietojen tyyppiä kuvaaviksi lisätiedoiksi. Lastausalueella tämä jaottelu konkretisoitui kahtena erillisenä tietokannan tauluna, joista ensimmäiseen tallennettiin ainoastaan tapahtumatiedot ja toiseen vastaavasti tapahtumien tyyppiä kuvaava lisäinformaatio.

Dataputken viimeisessä vaiheessa prosessointisilmukassa lastausalueen tietokantataulujen uusittu data tarkistettiin ensin mahdollisten duplikaattiavainten varalta. Jos tauluista löydettiin duplikaattitietueita, ne siirrettiin erilliseen virhetauluun mahdollisia jatkotarkistuksia varten. Tämänkaltaisella putken sisäänrakennetulla virneenhallintaprosessilla pystyttiin välttämään datalatauksen kaatuminen tilanteessa, jossa tunnistettu data-virhe haluttiin tallentaa kaatamatta kuitenkin koko datalatauksen suoritusta. Tarkastuksen läpi selvinneet puhdistetut tietueet parsittiin tietokantojen päälle luotujen

valmiiden kyselyjen eli tietokantanäkymien avulla strukturoidumpaan muotoon, joista ne lopulta päivitettiin varsinaisiin data-analytiikan käytössä oleviin tietokantatauluihin.

Dataputken kaikkia silmukoita ohjattiin Azure Data Factoryn avulla, jossa putken suoritusta pystyttiin seuraamaan palveluun sisäänrakennetun monitorointitoiminnon avulla. Lisäksi Data Factoryn sisältämää pilvipohjaista laskentatehoa hyödynnettiin lähdejärjestelmän raakadatan keräämisessä tietoaaltaaseen. Valitettavasti saman laskentatehon hyödyntäminen koettiin ongelmalliseksi tietokantalatauksien kohdalla, jos tietokantapalvelin konfiguroitiin päästämään lävitse ainoastaan tietyistä IP-osoitteista tuleva tietoliikenne. Data Factoryn käyttämät IP-osoitteet vaihtelevat tuotteen pilvipohjaisuuden vuoksi, jonka seurauksena niiden automaattinen hallinnoiminen olisi vaatinut erillisen automaatiologiikan kehittämistä käytetyn tietokannan yhteydessä. Tätä haluttiin välttää, ja lopulta tutkielmassa päädyttiin tietokantayhteyden osalta hyödyntämään erillistä virtuaalikonetta, jonka kautta tietokantalataukset sekä tietokantaan liittyvät muut toiminnot pystyttiin suorittamaan joustavammin ennalta tiedossa olevan staattisen IP-osoitteen ansiosta. Virtuaalikone liitettiin osaksi Data Factorya, jonka kautta tietokantaan liittyvät toiminnot pystyttiin ajamaan lävitse.

Varsinaisen datalatauksen lisäksi kuviossa 7 on kuvattu putken suoritusmahdollisuutta kahdessa erillisessä ympäristössä, tarkemmin kehitys- ja testausympäristöissä. Kuvion mukaisesti putki rakennettiin tavalla, jossa kehitysympäristössä luodut muutokset pystyttiin siirtämään erillisiin vain testausympäristön käytössä oleviin resursseihin. Tämä toiminnallisuus toteutettiin Azure Pipelines palveluun rakennetun prosessin avulla alla olevan kuvion 8 mukaisesti:



Kuvio 8. Suunnitellun dataputken jatkuva integraatio sekä toimitus kehitysympäristöstä testausympäristöön.

Jatkuvan integroinnin ja toimituksen prosessi rakennettiin kuvion 8 mukaisesti osaksi dataputken kehitysprosessia. Tämä toteutettiin luomalla erilliset Data Factory sekä tietokantaresurssit, joista toiset toimivat puhtaasti kehitykseen tarkoitettuna kehitysympäristönä ja toiset vastaavasti testaukseen soveltuvana testausympäristönä. Kehitysympäristössä tehdyt muutokset pystyttiin siten siirtämään testausympäristöön Azure Pipelines palveluun luodun integrointi- ja toimitusprosessin avulla. Integrointiprosessi oli mahdollista käynnistää siirtämällä kehitysympäristössä uusittu toiminnallisuus Azure DevOpsiin sisällytetyn versionhallinnan julkaisua seuraavaan osaan eli haaraan (englanniksi branch). Tämä siirto käynnisti automatisoidun julkaisuprosessin, jossa ensin integraatiovaiheessa versionhallinnan julkaisuhaaraan pohjautuen uusittu koodi ja toiminnallisuus niin Data Factoryn kuin myös tietokantaobjektien osalta integroitiin yhtenäiseksi julkaisupaketiksi (kuviossa 8 nimellä Artefakti). Integroinnin jälkeen prosessin julkaisuvaiheessa paketti julkaistiin testausympäristöön, jossa Data Factory ja tietokantaobjektit päivitettiin vastaamaan uusittua toiminnallisuutta. Lisäksi integraatio- ja julkaisuprosessiin kehitettiin yksinkertainen integraatiotesti, joka suoritettiin viimeisenä vaiheena onnistuneen julkaisun lopuksi testausympäristössä. Testillä käynnistettiin automatisoidusti Data Factoryn

sisältämä dataputken ajosuoritus, jolloin uusitusta putken toiminnallisuudesta pystyttiin varmistumaan onnistuneella testisuorituksella.

4.7 Arviointi

Artefaktin tavoitteeksi asetettiin toimia geneerisenä ylätasoina tukemaan DataOpsin periaatteita noudattavan dataputken suunnittelua ja kehitystä, jonka avulla pystyttäisiin vastaamaan data-analytiikan kohtaamiin haasteisiin. Lisäksi tutkielman konkreettisenä tavoitteena oli tuottaa toimiva esimerkki DataOpsin käytänteitä noudattavasta dataputkesta. Jälkimmäistä tavoitetta voidaan arvioida vertaamalla, kuinka tutkielman demonstraatiovaiheessa artefaktin avulla kehitetyn dataputken ominaisuudet vastaavat artefaktissa listattuja ominaisuuksia sekä niihin liitettyjä kysymyksiä.

Toistettavuus.

Suunniteltu dataputki on toistettavissa toisessa ympäristössä, joka osoitettiin demonstrointivaiheessa, missä dataputki sekä siihen liitettyt uudistetut muutokset siirrettiin kehitysympäristöstä testausympäristöön. Lisäksi dataputken suoritus voidaan toistaa tarvittaessa, joko ajamalla putki läpi kokonaisuudessaan, tai vaihtoehtoisesti käynnistämällä ainoastaan tietoaaltaan jälkeen tapahtuva lataamis- ja muokausprosessi. Tämä putken uudelleenkäynnistys mahdollistettiin jakamalla dataputki kahteen erilliseen silmukkaan Data Factoryn sisällä, joista ensimmäinen silmukka sisälsi keräysvaiheen toiminnot tietoaaltaaseen, jälkimmäisen sisältäessä vastaavasti datalatauksen ja esikäsittelyn sekä varsinaisen prosessoinnin tietokantaan asti. Kaikkien silmukoiden suoritus sisältyi kuitenkin yhteen ylemmän tason dataputkeen, joka käynnistettiin Data Factoryn avulla (kuviossa 7 lataus ja esikäsittely sekä prosessointi ovat esitetty omina erillisinä silmukoina kuvion selkeyttämisen vuoksi). Dataputken voidaan siis katsoa olevan toistettavissa.

Ajoympäristö on myös toistettavissa, jota demonstroitiin luomalla identtiset resurssit sekä kehitys- että testausympäristöön. Toisaalta ajoympäristön siirtäminen sellaisenaan toiseen ympäristöön ei suoraan ole mahdollista vaan simuloitu toteutus vaatii manuaalista työtä, jos kehitys- tai testausympäristö halutaan siirtää muualle. Ajoympäristö on siten vain osittain toistettavissa.

Automatisoitavuus.

Demonstroitu dataputki sekä siihen liittyvä kehitystyö ovat automatisoitavissa, josta osoituksena jatkuvan integroinnin sekä toimituksen liittäminen osaksi simuloitua dataputken prosessia. Lisäksi integrointiprosessi käynnistyy automaattisesti, kun kehityksessä uusittu toiminnallisuus siirretään versionhallinnan julkaisua seuraavaan osuuteen eli haaraan. Demonstroidun dataputken ajo ei ole automatisoitu, tosin se on mahdollista toteuttaa tarvittaessa luomalla ajastettu käynnistys Data Factoryn toiminnallisuuksia hyödyntäen.

Kehitys- tai testausympäristön luominen ei sen sijaan ole automatisoitavissa. Tämä on selkeä puute demonstroidussa dataputkessa, ja sen suunnittelu tulisi ottaa huomioon seuraavalla iteraatiokierroksella.

Testattavuus.

Suunniteltu dataputki on testattavissa testausympäristössä, joka osoitettiin sisällyttämällä dataputken ajamisen suoritus testausympäristön integraatiotestiin. Lisäksi dataputkeen rakennettiin toiminnallisuus, joka siirsi mahdolliset lähdedatasta löydetyt duplikaattitietueet erilliseen virhetauluun tietokannassa. Testi toteutettiin tutkimalla duplikaatteja perustuen lähdedatan tietueen avaimena toimineeseen kenttään.

Ajoympäristön testaaminen toteutettiin edellä mainitun integraatiotestin avulla, jossa ympäristön toimivuus varmistettiin ajamalla dataputki läpi testausympäristössä. Ajoympäristön toimivuus on siten myös testattavissa.

Seurattavuus.

Dataputken ajoa voidaan seurata Data Factorysta löytyvän monitorointitoiminnon avulla. Dataputki on siten seurattavissa.

Ajoympäristön tilaa pystytään seuraamaan Azuresta sen resursseihin sisältyvien monitorointitoimintojen avulla. Tosin yhtä yhtenäistä paikkaa, josta esimerkiksi sekä dataputken ajoa että ajoympäristön tilaa voisi seurata ei luotu tutkimuksen aikana. Tätä kehitysmahdollisuutta voisi tarkastella seuraavalla iteraatiokierroksella.

Tutkielman toisena tavoitteena artefaktin haluttiin toimivan geneerisenä ylätasoina lisätauksena, jonka avulla suunniteltu DataOpsin parhaita käytänteitä mukaileva dataputki pystyisi vastaamaan Data-analyysin ongelmat -kappaleessa esitettyihin Benghiatin ja muiden (2019, s. 8–12) esittämiin data-analyysin kohtaamiin haasteisiin. Tätä tavoitetta voidaan arvioida vertaamalla demonstrointivaiheessa simuloidun dataputken soveltuvuutta näiden ongelmakohtien ratkaisuksi. Alla olevaan taulukkoon 2 on koostettu yhteenvetona tämä vertailu:

Taulukko 2. Yhteenveto demonstroidun dataputken soveltuvuudesta Benghiatin ja muiden lisätaamiin data-analytiikan haasteisiin (2019, s. 11–12).

Haaste	Dataputki tarjoaa ratkaisun?
Jatkuvasti muuttuvat vaatimukset	Kyllä Jatkuva integrointi sekä toimitus sisältyvät dataputken kehitysprosessiin.
Datan eläminen siiloissa	Kyllä

	Dataa pystytään keräämään eri lähteistä luomalla uusia yhteyksiä Data Factoryssa sekä kehittämällä uusia dataputkia.
Datan epäsopiva muoto analytiikalle	Osittain Dataa pystytään muokkaamaan analytiikalle sopivaan muotoon dataputken silmukoiden avulla. Analyysien tekeminen jäi tutkimuksen ulkopuolelle eli tarvitsee vielä lisätutkimusta.
Datavirheet	Osittain Dataputkeen sisältyy testausmahdollisuus datavirheiden osalta. Toisaalta testaustapaus rajoittuu yksittäiseen virhetyyppiin eli testausta tulisi jatkokehittää.
Virheellinen raportointi	Osittain Dataputkeen sisältyy testausmahdollisuus. Analyysien tekeminen jäi tutkimuksen ulkopuolelle eli tarvitsee vielä lisätutkimusta.
Dataputkien jatkuva ylläpito	Kyllä Dataputkiin pystytään tekemään muutoksia kehitysprosessiin sisällytetyn jatkuvan integroinnin sekä toimituksen avulla. Testausmahdollisuus auttaa varmistamaan muutosten toimivan läpiviennin.
Manuaalinen työ	Osittain Dataputki sekä siihen liittyvä kehitysprosessi on pitkälti automatisoitu. Ajoympäristön siirtäminen sekä luomisprosessi vaatii sen sijaan manuaalista työtä.

”Toivon ja sankaruuden ansa”	Osittain Dataputken automatisoitu kehitysprosessi sekä testausmahdollisuus auttavat nopeuttamaan raportointia sekä laadunvarmistusta. Dataputkeen luotu testaus on kuitenkin erittäin minimaalinen eikä tarpeeksi kattava.
------------------------------	--

Yhteenvedona voidaan todeta, jotta demonstroitu dataputki noudattelee artefaktin listauksia geneerisiä ominaisuuksia. Dataputkea pystytään siis kuvailemaan toistettavaksi, automatisoitavaksi, testattavaksi sekä ainakin osittain seurattavaksi DataOpsin periaatteita mukailevaksi putkeksi. Suurimmat puutteet ovat nähtävissä ajoympäristöön liittyvissä ominaisuuksissa, joista etenkin automatisoitu ympäristön luominen jäi puutteelliseksi. Lisäksi testaaminen on minimaalista eli testitapausten kattavuutta tulisi lisätä huomattavasti. Artefaktin voidaan silti nähdä toteuttavan tehtävänsä, sillä nämä ongelmat nousivat selkeästi esille, ja niihin voidaan kiinnittää huomiota seuraavalla iteraatiokierroksella. Artefaktin ominaisuuksiin liitetyt kysymykset auttoivat suunnitteluvaiheen aikana, mutta ne eivät kata kaikkia mahdollisia tapauksia. Kysymysten tarkentamista, soveltuvuutta sekä täysin uusien kysymysten muodostamista voisi myös tutkia seuraavalla iteraatiokierroksella.

Demonstroitu dataputki vastasi suhteellisen hyvin myös Benghiatin ja muiden (2019, s. 11–12) esittämään listaukseen data-analytiikan kohtaamista haasteista, kuten taulukon 2 yhteenvedosta voidaan nähdä. Tämä tosin oli odotettavissa, sillä dataputken suunnittelun apuna käytetyn artefaktin tietämyspohja perustui vahvasti aikaisempaan tutkimustietoon DataOpsin määritelmästä sekä siihen liittyvistä parhaista käytänteistä, jotka ovat muodostettu ratkaisemaan juuri kyseisen listauksen esittämiä haasteita data-analytiikan kohdalla. Koska tämä tutkimus keskittyi enimmäkseen DataOpsin teknologiseen lähtökohtaan, ei tutkimuksen avulla voida vastata kysymykseen artefaktin soveltuvuudesta ihmisten tai ympäristön suhteen TOE-viitekehyksen ja DSRM-mallin esittämällä tavoilla.

Tämä on selkeä puute tutkielmalle, ja se voidaan nähdä erinomaisena jatkotutkimuskohteenä tulevaisuudessa.

4.8 Viestintä

Tutkimuksen lopputulokset esitellään kohdeyleisölle Vaasan yliopiston seminaarissa. Lisäksi tutkimus julkaistaan tiedeyhteisölle pro gradu -tutkielman muodossa.

5 Diskussio

Tutkielman tavoitteena oli tutustua DataOpsin määritelmään sekä sen tuomiin hyötyihin data-analyysin tuottamisessa. Tutkimuksen lopputuloksena luotiin suunnittelutieteellisen tutkimuksen DSRM-mallia hyödyntäen artefakti, joka muodostui listaukseksi tyypillisimmistä DataOpsin ominaisuuspiirteistä, jotka tulisi huomioida data-analytiikkaan läheisesti liittyvän datan siirtämisen, muokkaamisen ja lataamisen prosessissa eli dataputken suunnittelussa. Tämän artefaktin käyttöä demonstroitiin suunnittelemalla sen avulla DataOpsin parhaita käytänteitä mukaileva dataputki, jonka kehitystä sekä suoritusta simuloitiin Azuren pilviympäristössä. Tutkimuksen tavoitteeseen liittyvä tutkimuskysymys oli seuraava:

- Millainen on DataOpsin periaatteita noudattava datan siirtämiseen, muokkaamiseen ja lataamiseen kohdistuva prosessi?

5.1 Tulokset ja johtopäätökset

Tutkielman teoreettinen viitekehys muodostui kirjallisuudesta löytyvien erilaisten DataOpsin määritelmien pohjalle, joiden havaittiin eroavan toisistaan etenkin tarkasteltavan lähtökohdan suhteen. Data-analytiikan kohdalla osa määritelmistä painotti enemmän teknologisia ohjeistuksia, jolloin DataOpsin katsottiin lainaavan parhaita käytänteitä niin ketterän ohjelmistokehityksen (Ereth, 2018, s. 5), DevOpsin (Capizzin ja muut, 2019, s. 1–5; Sahoo ja muut, 2019, s. 1–2) kuin myös lean-ajattelun (Benghiat ja muut, 2019, s. 22–23) teorioista. Toisaalta osa löydetyistä määritelmistä kuvaili termiä enemmän organisatorisiin tekijöihin viitaten, kuten tavoittelemalla kommunikaation ja yhteistyön parantamista eri sidosryhmien välillä (Liebmann, 2014; Palmer, 2015). Myös Munappy ja muut (2020, s. 169) huomasivat tutkimuksessaan eri käyttäjäryhmien kuvailevan DataOpsia joko siihen sisältyvien aktiviteettien ja teknologioiden kautta tai enemmän sen avulla saavutettaviin lopputavoitteisiin ja organisaation sidosryhmien välisiin rakenteisiin perustuen. Teoreettisen viitekehysten perusteella DataOpsin määritelmälle ei ainakaan vielä ole löydettävissä yhtä yleisesti hyväksyttyä määritelmää. Toisaalta DataOps on

suhteellisen uusi tutkimuskohde tieteellisessä kirjallisuudessa, joka osaltaan vaikuttanee yksiselitteisen määritelmän puuttumiseen.

Vaihtelevista kuvauksista huolimatta kaikki löydetty määritelmät mukailivat toisiaan eikä niiden välillä nähty selkeitä ristiriitoja. Liebmannin (2014) esittämän ajatuksen mukaisesti DataOpsin tavoitteena on parantaa kommunikaatiota datatieteilijöiden sekä tuotannon prosessien välillä, johon liittyy niin teknologisia toteutuksia kuin myös ihmisiin ja organisaation kulttuuriin vaikuttavia tekijöitä. Erethin (2018, s. 5–6) ehdottama TOE-vii-tekeykseen pohjautuva malli DataOpsin tutkimukselle tukee tätä näkemystä, jolloin tutkimuksen lähtökohtana voidaan painottaa joko teknologisia ratkaisuja, ympäristöön liittyviä osa-alueita tai vastaavasti organisatorisia näkökulmia. Työkalujen ja ohjeistuksen lisäksi DataOps voidaan siis mieltää dataan liittyväksi strategiaksi, joka luo pohjan muutokselle dataa hyödyntävissä yrityksissä (Munappy ja muut, 2020, s. 169). Organisaatioiden kohtaamat haasteet data-analytiikan investointien muuntamisessa lisäarvoa tuottaviksi tekijöiksi (ks. NewVantage Partners, 2019; PwC, 2015) sekä itse analytiikan kohtaamat ongelmakohdat (ks. Benghiat ja muut, 2019, s. 8–12) pakottavat yrityksiä etsimään ratkaisuja, joihin DataOpsin parhaat käytänteet sekä DataOps manifeston listaamat periaatteet (DataOps manifesto, 2021) voivat toimia erinomaisena lähtökohtana.

Tutkielman teoriakatsauksen perusteella DataOps voidaan nähdä olevan joukko DevOpsiin, ketterään kehittämiseen sekä lean-ajatteluun pohjautuvia parhaita käytänteitä, prosesseja sekä menetelmiä, joiden avulla datan laatua sekä sen analysointia voidaan parantaa ja nopeuttaa, edistäen samalla jatkuvan kehittämisen kulttuuria sekä yhteistyötä eri sidosryhmien välillä. Lisäksi katsauksen avulla löydettiin ainakin kolme päävaihetta, joita seuraamalla DataOps pystytään implementoimaan teknologisenä toteutuksena. Nämä data-analytiikan kehityksessä hyödynnettävät vaiheet olivat rakennus, suoritus sekä käyttö (Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Yhdessä ne muodostavat jatkuvan kierteen eli DataOpsin kehitys- ja julkaisuprosessin, jossa vaiheita toistetaan uudelleen samassa järjestyksessä analytiikkaan liittyvässä kehitystyössä (Benghiat ja muut, 2019, s. 37). Näitä vaiheita mukailtiin myös tässä tutkielmassa soveltuvien osien.

Suunnittelutieteellisellä tutkimuksella ja sen avuksi kehitetyn DSRM-mallin avulla tutkielmassa suunniteltiin artefakti, joka muodostui listaukseksi DataOpsin tyypillisimmistä ominaisuuspiirteistä, kun tavoitteena on luoda määritelmän parhaita käytänteitä mukailtava datan siirtämisen, muokkaamisen ja lataamisen prosessi eli dataputki. Listauksessa nostettiin esille neljä tyypillisintä piirrettä kuvaamaan DataOpsin periaatteita noudattavan dataputken suunnittelua, jotka nähtiin seuraaviksi: toistettavuus, automatisoitavuus, testattavuus ja seurattavuus. Dataputken ominaisuutena toistettavuus kuvastaa putken kyvykkyyttä siirtyä toiseen ajoympäristöön mahdollistaen sen suorittamisen eli toistamisen uudessa ympäristössä. Lisäksi dataputken sisältämät aktiviteetit olisi hyvä suunnitella siten, että niiden uudelleenkäynnistys on mahdollista esimerkiksi virhetilanteen satuttaessa (vrt. Raj ja muut, 2020, s. 17–18). Ajoympäristön osalta toistettavuudella vastavasti kuvataan mahdollisuutta toistaa tai siirtää ympäristön tarvitsemat resurssit identtisinä uuteen ympäristöön, jotta dataputki pystytään siirtämään ja toistamaan edellä mainitulla tavalla myös tässä ympäristössä.

Toiseksi ominaisuudeksi nähtiin toistettavuuteen läheisesti liittyvä automatisoitavuus, jota korostetaan laajasti DataOpsin eri kuvauksissa (ks. Benghiat ja muut, 2019, s. 22; Capizzi ja muut, 2019, s. 1–5; Ereth, 2018, s. 5; Munappy ja muut, 2020, s. 169; Palmer, 2015; Sahoo ja muut, 2019, s. 1–2). Automatisoitavuudella tarkoitetaan dataputken sekä sen ajoympäristön resurssien kyvykkyyttä olla automaattisesti siirrettävissä tai luotavissa eri ympäristöjen välillä. Parhaimmassa tapauksessa dataputki sekä sen tarvitsemat resurssit ovat toistettavissa siten, jotta dataputken luominen alusta alkaen mahdollistetaan automatisoidulla tavalla. Artefaktin kolmanneksi piirteeksi muodostui testattavuus, jonka ajatuksena sekä dataa että dataputken suoritukseen liittyviä toimintoja tulisi pystyä testaamaan, jotta niiden oikeanlaisesta toimivuudesta eri ympäristöissä voitaisiin varmistua. Etenkin Benghiat ja muut (2019, s. 20–22, 25–26) painottivat testaamisen tärkeyttä myös mahdollisten datavirheiden kohdalla logiikkaan kohdistuvien testitapausten lisäksi.

Viimeiseksi artefaktin piirteeksi nousi seurattavuus, joka viittaa dataputken suorituksen sekä siihen liittyvien ympäristön resurssien monitoroimiseen, jotta mahdollisiin ongelmatilanteisiin pystytään reagoimaan (vrt. Benghiat ja muut, 2019, s. 20–22; Capizzi ja muut, 2019, s. 4–5; Patterson, 2018). Tilastollisen prosessikontrollin menetelmät ovat varteenotettavia keinoja tämän toimivuuden varmistamiseen, kuten Benghiat ja muut (2019, s. 20–22) esittävät.

Näiden neljän ominaisuuspiirteen sisältävän artefaktin käyttöä demonstroitiin suunnitteleamalla sekä kehittämällä dataputki Azuren pilviympäristöön sen sisältämiä palveluita ja resursseja hyödyntämällä. Putki onnistuttiin luomaan artefaktin avulla DataOpsin piirteitä mukailevaksi datan siirtämiseen, muokkaamiseen ja lataamiseen tarkoitetuksi prosessiksi. Dataputken todettiin olevan toistettavissa toisessa ympäristössä, automatisoitu kehitysprosessin osalta, testattavissa niin datavirheen kuin myös ajoympäristön suhteen sekä seurattavissa Data Factoryyn sisäänrakennetun monitorointitoiminnon avulla. Toisaalta prosessista ei onnistuttu luomaan täysin automaattista, ja varsinkin ajoympäristön automatisoitavuuden parantamista sekä testitapausten lisäämistä tulisi tarkastella uudella iteraatiokierroksella. Nämä puutteet kuitenkin huomattiin, kun dataputkea verrattiin artefaktin sisältämään listaukseen tarvittavista ominaisuuksista. Tämä myös kuvastaa sen tarkoitusta toimia enemmän dataputken suunnittelua ja kehitystä suuntaavana tekijänä kohti DataOpsin sisältämiä ajatuksia. Artefakti on siten työkalu ohjaamaan kehittäjiä oikeanlaiseen suuntaan jokapäiväisessä kehitystyössä, eikä sen avulla pystytä täysin varmistamaan DataOpsin käytänteitä noudattavan dataputken toteuttamista. Tämä on myös tärkeää huomioida artefaktia käytettäessä.

Tutkimuksen teoreettisen viitekehyksen sekä siihen perustuvan suunnittelutieteellisen artefaktin avulla tutkimuksen tutkimuskysymykseen voidaan vastata kuvailemalla DataOpsin periaatteita noudattavan datan siirtämisen, muokkaamisen ja lataamisen prosessi olevan toistettavissa, automatisoitavissa, testattavissa sekä seurattavissa oleva datan käsittelyyn hyödynnettävä dataputki. Tutkimuksessa luodun artefaktin avulla ei välttämättä onnistuttu luomaan uutta tietoa DataOpsin määritelmään tai teoriaan liittyen, sillä

artefakti pohjautuu vahvasti jo aikaisempaan tutkimustietoon. Tutkielman isoimpana kontribuutiona voidaan siten pitää tutkimuksen aihealueen eli DataOpsin esittelyä, sekä kuinka sen parhaita käytänteitä pystytään jalkauttamaan dataputken suunnittelu- ja kehitystyöhön teknologista näkökulmaa painottaen.

5.2 Rajoitteet ja tutkimuksen arviointi

Tutkielman suunnittelutieteellisen tutkimuksen avulla toteutettu artefakti keskittyi vastaamaan tutkimusongelmaan puhtaasti teknologista näkökulmaa painottaen. Tämä on myös tutkimuksen suurin rajoite, sillä suunnittelutieteellisen tutkimuksen DSRM-mallissa teknologisen lähtökohdan lisäksi toimivat organisaatio sekä ihmiset. Lopputuloksena luodun artefaktin toimivuudesta ei ole varmuutta, kun tarkastelun painopistettä kohdistetaan enemmän näitä kahta lähtökohtaa vasten. Sama rajoite kohdistuu myös DataOpsin määritelmään, joka rajattiin tässä tutkimuksessa kohdistumaan teknologisiin aihealueisiin. Kuten teoriakatsauksessa esitetyissä DataOpsin kuvauksissa huomataan, ei DataOps ole pelkästään teknologiaa painottava menetelmä, vaan se on tarkoitettu käytettäväksi laajemmin esimerkiksi yhteistyön ja organisaation kulttuuria muokkaavaksi ajatukseksi laadukkaamman ja toimivamman data-analyysin tuottamisen mahdollistajaksi. Ihmisten ja organisaation tarkastelun puuttuminen on siten selkeä rajoite tutkimukselle.

Toisena tutkimuksen rajoitteena voidaan nähdä varsin suppea tieteellinen tutkimustieto DataOpsin määritelmästä. DataOps on yritysmaailmassa viime vuosina yleistynyt ajatus paremman data-analyysin tuottajana mutta määritelmään liittyvä tieteellinen tutkimus on vielä varsin vähäistä. Tutkimuksen teoreettinen viitekehys on siksi muodostettu tieteellisen tutkimustiedon sekä osittain yleisemmin saatavilla olevan kirjallisuuden pohjalta, joka on rajoite tutkielmalle. Viitteitä DataOpsin tieteellisen tutkimuksen kasvusta on kuitenkin havaittavissa, ja myös tämä työ voidaan nähdä yhtenä kontribuutiona aihealueen tutkimukselle.

Alla olevassa taulukossa 3 tutkielman käytännön hyötyjä on arvioitu Hevnerin ja muiden (2004, s. 82–90) suunnittelutieteellisen tutkimuksen ohjeisiin verraten:

Taulukko 3. Tutkielman arviointi suunnittelutieteellisen tutkimuksen ohjeistukseen verraten.

Ohje	Tutkielman arviointi
1. Suunnittelutieteellisen tutkimuksen tulee tuottaa käyttökelpoinen artefakti rakenteellisessa muodossa, mallissa, metodissa tai ilmentymänä.	Tutkielmassa luotiin artefakti, joka muodostui listaukseksi tyypillisimmistä ominaisuuksista, joita DataOpsin periaatteita noudattava datan siirtämisen, muokkaamisen ja lataamisen prosessi eli dataputki pitää sisällään. Artefakti muodostettiin DSRM-mallin avulla ja se pohjautuu vahvasti jo aikaisempaan tutkimustietoon. Listauksen käyttöä demonstroitiin kehittämällä sen avulla dataputki Azuren pilviympäristöön.
2. Suunnittelutieteellisen tutkimuksen tavoitteena on kehittää teknologiaan pohjautuvia ratkaisuja tärkeisiin ja asiaankuuluviin liiketoiminnan ongelmiin.	Tutkielman tavoitteena oli tuottaa yleiskäyttöinen listaus, jota hyödyntämällä data-analytiikan kehittäjät pystyisivät luomaan dataputken DataOpsin periaatteiden mukaisesti. Dataputken ominaisuudet ovat ylätasolla erittäin toistuvia, jolloin listauksen kysymykset huomioimalla kehittäjät pystyvät nopeasti varmistamaan suunnitellun dataputken toteuttavan ainakin osittain kaikki neljä tarvittavaa ominaisuutta. Artefakti auttaa kehittäjiä kohti DataOpsin teknologisten periaatteiden käyttöönottoa mutta ei kuitenkaan takaa sen toteutumista.

3. Artefaktin hyödyllisyys, laatu ja tehokkuus on osoitettava hyvin toteutettujen arviointimenetelmien avulla.	Artefaktin hyödyllisyys osoitettiin tutkielman demonstraatiovaiheessa, jossa artefaktin käyttöä simuloitiin kehittämällä DataOpsin periaatteita mukaileva dataputki. Lisäksi suunnitellun putken tarjoamia hyötyjä verrattiin Benghiatin ja muiden (2019, s. 8–12) esittämiin data-analytiikan kohtaamiin haasteisiin. Simuloitu dataputki ratkaisi osittain nämä ongelmakohdat.
4. Tehokkaan suunnittelutieteellisen tutkimuksen on tuotettava selkeää ja todennettavissa olevaa kontribuutiota suunniteltuun artefaktiin, suunnittelun perusteisiin ja/tai metodologeihin.	Tutkimuksen artefakti tarjoaa yhteenvetona ylätasoa listauksen ominaisuuksista, joita DataOpsin periaatteita noudattava dataputki pitää sisällään. Listaus auttaa kehittäjiä näiden ominaisuuksien huomioimisessa dataputken suunnittelu- ja kehitysvaiheessa. Lisäksi tutkielma lisää yleistä tietoisuutta DataOpsista, jonka tieteellinen tutkimus on vielä vähäistä.
5. Suunnittelutieteellinen tutkimus perustuu täsmällisiin metodeihin artefaktin rakennus- ja arviointivaiheissa.	Tutkielma pohjautuu vahvasti jo aikaisemman saatavilla olevaan tutkimustietoon sekä yleiseen kirjallisuuteen. Lisäksi tutkimus hyödyntää suunnittelutieteellisen tutkimuksen DSRM-mallia artefaktin suunnittelussa.
6. Tehokkaan artefaktin etsiminen edellyttää käytettävissä olevien keinojen hyödyntämistä tyydyttävän ratkaisun löytämiseksi,	Artefaktin sisältämät ominaisuudet ovat muodostettu DataOpsin tutkimuksen sekä yleisen kirjallisuuden pohjalta. Lisäksi listauksen käyttöä demonstroitiin Azuren pilviympäristössä, jossa sen

huomioimalla kuitenkin ympäristön asettamat lainalaisuudet.	tarjoamia resursseja hyödynnettiin ympäristön asettamia lainalaisuuksia soveltaen.
7. Suunnittelutieteellinen tutkimus on esitettävä tehokkaasti sekä teknologian asiantuntijoille kuin myös yritysjohdolle.	Tutkimus esitettiin soveltuvalle kohdeyleisölle Vaasan yliopistossa. Lisäksi tutkielma julkaistaan pro gradu -tutkielmana tutkijayhteisön käyttöön.

5.3 Jatkotutkimusaiheet

Tutkimuksen artefaktin keskittyessä ainoastaan teknologiseen näkökulmaan, tulisi sen soveltuvuutta tutkia suunnittelutieteellisen tutkimuksen mukaisesti myös organisaation ja ihmisten suhteen. Näiden aihealueiden huomioimista painotetaan myös DataOpsin tieteellisessä tutkimuksessa teknologisen näkökulman lisäksi, jolloin artefaktin avulla suunnitellun dataputken toimivuutta voisi tutkia tuomalla esimerkiksi sidosryhmien kanssa toteutettavan vaatimusmäärittelyn osaksi putken kehitystä. Tällöin artefaktin toimivuutta voitaisiin tutkia enemmän data-analyysin raporttien tuottamisen kannalta.

Toisena jatkotutkimuskohteena tutkimuksessa demonstroitua dataputkea tulisi kehittää tukemaan enemmän sen ajoympäristöön liittyvää automatisointia. Lisäksi seurattavuutta voisi edistää rakentamalla yhtenäinen järjestelmä, jonka kautta sekä putkea että ajoympäristöön liittyviä resursseja voisi monitoroida. Tällä hetkellä monitorointi tapahtuu resurssien sisällä eikä tämä ole skaalautuva prosessi. Myös dataputkeen liittyvää testausta tulisi kehittää kattavammaksi, jotta pystyttäisiin varmistamaan laadukas raportointi. Demonstroitua putkea voisi testata myös tuotantokäytössä, jossa esimerkiksi tietoturvaan vaikuttavat tekijät tulisi huomioida datan käsittelyn aikana. Itse artefaktin sisältämiä kysymyksiä voisi myös tarkentaa sekä mahdollisesti luoda täysin uusia versioita tukemaan paremmin onnistunutta DataOpsin periaatteita mukailevan dataputken kehitystä.

Kolmantena jatkotutkimusaiheena voidaan pitää yleisesti DataOpsiin liittyvää tieteellistä tutkimusta sekä sen määritelmän tarkentamista. Nopeasti tuotettavan ja luotettavan data-analytiikan muodostaminen on nykypäivän yrityksille merkittävä kilpailutekijä, jolloin esimerkiksi case-tutkimuksen avulla voitaisiin tutkia, toteutuuko DataOpsin lupaamat tavoitteet organisaatiossa DataOpsin ajatuksia hyödyntämällä.

Lähteet

- Agile Manifesto (2021). Manifesto for Agile Software Development. Noudettu 5.12.2021 osoitteesta <https://agilemanifesto.org/>
- ASQ (2021). What Is Statistical Process Control?. Noudettu 7.11.2021 osoitteesta <https://asq.org/quality-resources/statistical-process-control>
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development. Noudettu 7.11.2021 osoitteesta <https://agilemanifesto.org/>
- Balachandran, B. M. & Prasad, S. (2017). Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business Intelligence. *Procedia Computer Science*, 112, 1112–1122. <https://doi.org/10.1016/j.procs.2017.08.138>
- Benghiat, G., Bergh, C. & Strod, E. (2019). *The DataOps Cookbook*. DataKitchen, Inc.
- Capizzi, A., Distefano, S., Mazzara, M. (2020) From DevOps to DevDataOps: Data Management in DevOps Processes. *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*. https://doi.org/10.1007/978-3-030-39306-9_4
- Chaudhuri, S. & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1), 65–74. <https://doi.org/10.1145/248603.248616>
- DataOps Manifesto (2021). The DataOpsManifesto. Noudettu 5.12.2021 osoitteesta <https://dataopsmanifesto.org/en/>
- Dingsøyr, T., Nerur, S., Balijepally, V. & Moe, N., B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. <https://doi.org/10.1016/j.jss.2012.02.033>
- Docker (2021) Use containers to Build, Share and Run your applications. Noudettu 30.10.2021 osoitteesta <https://www.docker.com/resources/what-container>
- Dyck, A., Penners, R. & Lichter, H. (2015). *Towards Definitions for Release Engineering and DevOps*. 2015 IEEE/ACM 3rd International Workshop on Release Engineering, 3–3. <https://doi.org/10.1109/RELENG.2015.10>

- Ereth, J. (2018). DataOps - Towards a Definition. Stuttgart, Saksa. Noudettu 14.3.2021 osoitteesta https://www.researchgate.net/publication/327542051_DataOps_-_Towards_a_Definition
- Flyway (2021). Documentation. Noudettu 30.10.2021 osoitteesta <https://flywaydb.org/documentation/>
- Gantz, J., Reinsel, D. & Rydning, J. (2018). *The Digitization of the World - From Edge to Core*. International Data Corporation. Noudettu 14.3.2021 osoitteesta <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>
- Haryono, E. M., Fahmi, Tri W, A. S., Gunawan, I., Hidayanto, A. N. & Rahardja, U. (2020) Comparison of the E-LT vs ETL Method in Data Warehouse Implementation: A Qualitative Study. *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 115–120. <https://doi.org/10.1109/ICIMCIS51567.2020.9354284>
- Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design Science in Information Systems. *Research. MIS Quarterly*, 28(1), 75–105.
- Kleppmann, M. (2017). *Designing Data-Intensive Applications*. Storage and Retrieval (s. 90–94). O'Reilly Media, Inc.
- Lennerholt, C., van Laere, J. & Söderström, E. (2018). Implementation Challenges of Self Service Business Intelligence: A Literature Review. *Hawaii International Conference on System Sciences*, 5055–5063. <https://dx.doi.org/10.24251/HICSS.2018.631>
- Liebmann, L. (2014, 19. kesäkuuta). 3 reasons why DataOps is essential for big data success. Noudettu 2020-03-05 osoitteesta <https://www.ibmbigdatahub.com/blog/3-reasons-why-dataops-essential-big-data-success>
- Mathis, C. (2017). Data Lakes. *Datenbank Spektrum*, 17, 289–293. <https://doi.org/10.1007/s13222-017-0272-7>
- Merchan, W. (2018, 30. tammikuuta). Why companies are failing at DataOps - and what

to do about it. Noudettu 2020-03-03 osoitteesta <https://www.cio-dive.com/news/why-companies-are-failing-at-dataops-and-what-to-do-about-it/515865/>

Microsoft (2021a). What is Azure. Noudettu 30.10.2021 osoitteesta <https://azure.microsoft.com/en-us/overview/what-is-azure/>

Microsoft (2021b). Introduction to Data Factory. Noudettu 30.10.2021 osoitteesta <https://docs.microsoft.com/en-us/azure/data-factory/introduction>

Microsoft (2021c). What is Azure DevOps. Noudettu 30.10.2021 osoitteesta <https://docs.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops>

Microsoft (2021d). Introduction to Blob Storage. Noudettu 30.10.2021 osoitteesta <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>

Microsoft (2021e). What is Azure DB for PostgreSQL. Noudettu 30.10.2021 osoitteesta <https://docs.microsoft.com/en-us/azure/postgresql/overview>

Munappy, A., R., Mattos, D., I., Bosch, J., Holmström Olsson, H. & Dakkak, A. (2020). From Ad-Hoc Data Analytics to DataOps. *Proceedings of the International Conference on Software and System Processes (ICSSP '20)*, 165–174. <https://doi.org/10.1145/3379177.3388909>

MyHelsinki (2021). MyHelsinki Open API v1. Noudettu 28.10.2021 osoitteesta <https://open-api.myhelsinki.fi/doc>

NewVantage Partners (2019). *Big Data and AI Executive Survey 2019*. NewVantage Partners LLC.

Palmer, A. (2015, 7. toukokuuta). From DevOps to DataOps. Noudettu 2020-03-05 osoitteesta <https://www.tamr.com/blog/from-devops-to-dataops-by-andy-palmer/>

Patterson, P. (2018, 24. toukokuuta). DataOps: Applying DevOps to Data - Continuous Dataflows Built With a DataOps Platform. Noudettu 2020-03-05 osoitteesta <https://dzone.com/articles/dataops-applying-devops-to-data-continuous-dataflo>

- Peppers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Pettersen, J. (2009). Defining lean production: some conceptual and practical issues. *TQM Journal*, 21(2), 127–142. <https://doi.org/10.1108/17542730910938137>
- PostgreSQL (2021). About PostgreSQL. Noudettu 30.10.2021 osoitteesta <https://www.postgresql.org/about/>
- PwC (2015). *Seizing the Information Advantage: How Organisations Can Unlock Value and Insight from the Information They Hold*. PricewaterhouseCoopers.
- Raj, A., Bosch, J., Olsson, H., H. & Wang, T., J. (2020). Modelling Data Pipelines. *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 13–20. <https://doi.org/10.1109/SEAA51224.2020.00014>
- Sahoo, P. R. & Premchand, A. (2019). DataOps in Manufacturing and Utilities Industries. *International Journal of Applied Information Systems*, 12(23), 1–6. <http://www.ijais.org/archives/volume12/number23/1060-2019451814>
- Stodder, D. (20. 12 2019). *Executive Summary for the TDWI Best Practices Report: Faster Insights from Faster Data*. Noudettu 14.3.2021 osoitteesta TDWI: <https://tdwi.org/articles/2019/12/20/data-all-executive-summary-bpr-faster-insights-from-faster-data.aspx>
- Teece, D., J., Pisano, G. & Shuen, A. (1998). Dynamic capabilities and strategic managements. *Strategic Management Journal*, 18(7), 509–533. [https://doi.org/10.1002/\(SICI\)1097-0266\(199708\)18:7%3C509::AID-SMJ882%3E3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1097-0266(199708)18:7%3C509::AID-SMJ882%3E3.0.CO;2-Z)

Liitteet

Liite 1. DataOps manifeston kahdeksantoista periaatetta (mukaillen ja kääntäen DataOps Manifesto, 2021).

1. Tyydytä jatkuvasti asiakkaasi.
 - a. Tärkein prioriteetti on tyydyttää asiakas tuottamalla arvokasta analyttistä tietoa jo varhaisessa vaiheessa ja jatkuvalla tahdilla, muutamien minuuttien tai muutamien viikkojen aikaväleillä.
2. Arvosta toimivaa analytiikkaa.
 - a. Toimivan data-analytiikan suorituskyvyn ensisijainen mittari on se, missä määrin oivaltavia analyyskejä tuotetaan edustavan ja tarkan datan avulla hyödyntämällä luotettavia viitekehyksiä sekä järjestelmiä.
3. Hyväksy muutos.
 - a. Toivota tervetulleeksi jatkuvasti kehittyvät asiakkaan vaatimukset, itse asiassa, kannusta luomaan kilpailullista etua. Kaikista tehokkain, vaikuttavin sekä ketterin kommunikointi asiakkaan kanssa tapahtuu kasvokkain käytävillä keskusteluilla.
4. Se on joukkueurheilua.
 - a. Analytiikkatiimi muodostuu aina yhdistelmästä erilaisia rooleja, taitoja, suosikkityökaluja sekä ammattinimikkeitä. Taustojen ja mielipiteiden moninaisuus lisää innovointia sekä tuottavuutta.
5. Päivittäiset vuorovaikutukset.
 - a. Asiakkaiden, analytiikkatiimien sekä toimintojen tulee työskennellä yhdessä päivittäin koko projektin elinkaaren ajan.
6. Itseorganisoidu.
 - a. Parhaimmat analyttiset tiedot, algoritmit, arkkitehtuurit, vaatimukset sekä mallit syntyvät itseorganisoituvista tiimeistä.
7. Vähennä sankaruutta.

- a. Analyttisen tiedon tarpeen nopeutuessa ja sen laajuuden jatkuvasti lisääntyessä, analytiikkatiimien tulisi pyrkiä vähentämään sankaruutta luomalla kestäviä sekä skaalautuvia data-analytiikkatiimejä sekä -prosesseja.

8. Reflektoi.

- a. Analytiikkatiimien tulisi hienosäätää heidän toiminnallista suorituskyykyään itsereflektoimalla tasaisin väliajoin, perustuen asiakkaalta, tiimiltä itseltään sekä toimintatilastoista saatavaan palautteeseen.

9. Analytiikka on koodia.

- a. Analytiikkatiimit käyttävät erilaisia yksittäisiä työkaluja datan saamiseen, integrointiin, mallintamiseen sekä visualisointiin. Periaatteessa, jokainen näistä työkaluista tuottaa koodia ja konfiguraatioita, jotka kuvailevat datalle tehtävät toiminnot, joilla tuotetaan tietoa.

10. Orkestroi.

- a. Datat, työkalujen, koodin, ympäristöjen ja analytiikkatiimien työn alusta loppuun asti ulottuva orkestrointi on analytiikan menestyksen avaintekijä.

11. Tee siitä toistettavissa oleva.

- a. Tarvitaan toistettavia tuloksia, ja siksi kaikki tulisi versioida: data, matalan tason laitteisto- ja ohjelmistokokoonpanot sekä kunkin työkaluketjun työkaluun liittyvä koodi ja kokoonpano.

12. Kertakäyttöiset ympäristöt.

- a. On tärkeää minimoida analytiikkatiimin jäsenten kokeilukustannukset tarjoamalla heille helposti luotavia, eristettyjä, turvallisia sekä kertakäyttöisiä teknisiä ympäristöjä, jotka vastaavat tuotantoympäristöjä.

13. Yksinkertaisuus.

- a. Jatkuva teknisen huippuosaamisen ja hyvän suunnittelun tarkastelu lisää ketteryyttä: samoin yksinkertaisuus – taito maksimoida tekemättömän työn määrä – on olennaista.

14. Analytiikka on valmistusta.

- a. Analytiikkaputket ovat vastaavanlaisia kuin lean-ajattelun valmistuslinjat. DataOpsin peruskäsitteenä on keskittyminen prosessiajatteluun, jolla

pyritään saavuttamaan jatkuvaa tehokkuutta analyyttisen tiedon valmistuksessa.

15. Laatu on ensisijaista.

- a. Analytiikkaputket tulisi rakentaa perustalle, joka kykenee havaitsemaan automaattisesti poikkeavuuksia ja tietoturvaongelmia koodissa, konfiguraatiossa ja datassa, sekä antamaan jatkuvaa palautetta toimijoille virheiden välttämiseksi.

16. Monitoroi laatua ja suorituskyykyä.

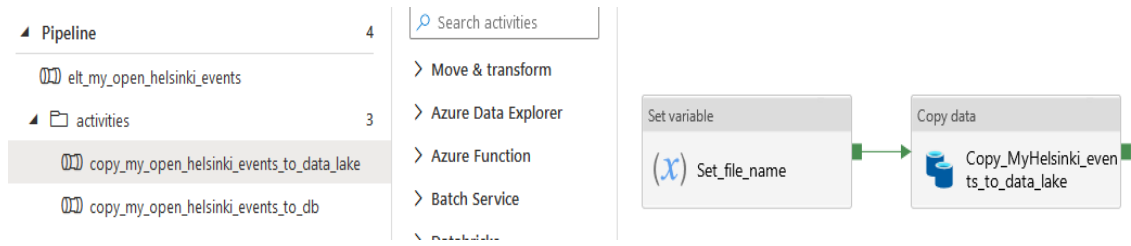
- a. Tavoitteena on suorituskyyky-, turvallisuus- ja laatutoimenpiteet, joita monitoroidaan jatkuvasti odottamattomien vaihteluiden havaitsemiseksi sekä toiminnallisten tilastojen tuottamiseksi.

17. Uudelleenkäyttö.

- a. Analyyttisen tiedon valmistuksen tehokkuuden perustavanlaatuinen näkökohta on välttää yksilön tai tiimin aikaisemman työn toistamista.

18. Paranna kiertoaikoja.

- a. Pyri minimoimaan aika ja vaiva, joka kuluu asiakkaan tarpeen muuttamisessa analyyttiseksi ideaksi, sen luomiseksi kehitystyössä, sen julkaisemiseksi toistettavaksi tuotantoprosessiksi, ja lopuksi refaktoroi ja uudelleen käytä tuotetta.

Liite 2. Kuvankaappaus Azure Data Factoryn dataputkesta.

Liite 3. Kuvankaappaus onnistuneesta jatkuvan integroinnin sekä toimituksen ajosta Azure Pipelines palvelussa.

←

Jobs in run #20211018.8

cicd-adf-pipeline-poc

Build Data Factory and database migration scripts

✓	✓	CreateADFAAndDbScripts	10s
	✓	Initialize job	1s
	✓	Checkout cicd-adf-pipeline-poc@a...	1s
	✓	Copy Azure Data Factory to: /ho...	<1s
	✓	Checkout cicd-adf-pipeline-poc@...	1s
	✓	Copy database scripts to: /home...	<1s
	✓	Copy pytest scripts to: /home/vs...	<1s
	✓	Publish artifact	4s
	✓	Post-job: Checkout cicd-adf-pipe...	<1s
	✓	Post-job: Checkout cicd-adf-pipe...	<1s
	✓	Finalize Job	<1s

Deploy to QA

✓	✓	QA_cicd_adf_poc_pipeline	3m 1s
	✓	Initialize job	4s
	✓	Download Artifact	4s
	✓	Deploy ADF from arm template	17s
	✓	Get agent machine IP address	<1s
	✓	Create temporary server firewall ...	22s
	✓	Install Docker	1s
	✓	Clean up QA database	14s
	✓	Deploy QA database	7s
	✓	Use python version 3.6	<1s
	✓	Install pytest dependencies	14s
	✓	Run pytest	1m 9s
	✓	Publish test results	4s
	✓	Remove temporary server firewal...	20s
	✓	Finalize Job	<1s

✓ QA_cicd_adf_poc_pipeline

1

Pool: [Azure Pipelines](#)

2

Image: ubuntu-latest

3

Agent: Hosted Agent

4

Started: Today at 7:17 PM

5

Duration: 3m 1s

6

7

100% tests passed

8

Job live console data:

9

Finishing: QA_cicd_adf_poc_pipeline