Juuso Issakainen

# DEVOPS – HISTORIA JA NYKYTILA

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2024

# TIIVISTELMÄ

Issakainen, Juuso
DevOps – Historia ja Nykytila
Jyväskylä: Jyväskylän yliopisto, 2024, 20 s.
Tietojärjestelmätiede, kandidaatintutkielma
Ohjaaja: Kyppö, Jorma

DevOps on kehittynyt viimeisten 16 vuoden aikana. Huolimatta sen laajasta käyttöönotosta, yleisesti hyväksyttyä määritelmää ei ole vielä selkeästi muodostettu. Tämä kirjallisuuskatsauksena toteutettu tutkimus syventyy ohjelmistotuotannon historiaan, DevOpsin määritelmiin, sisältöön, käytäntöihin ja haasteisiin. Lähdemateriaalina on käytetty vertaisarvioituja artikkeleita ja tutkimuksia sekä alan kirjallisuutta. Historiallinen konteksti kattaa siirtymisen vesiputousmallista ketteriin menetelmiin, korostaen niitä puutteita, joita DevOps pyrkii ratkaisemaan. Tutkimus ei löydä yhtä selkeää määritelmää DevOpsille. Tutkimus myös toteaa, että DevOps on kehittynyt vastauksena tarpeeseen parantaa ohjelmistokehityksen ja ylläpitotoimintojen välistä yhteistyötä. DevOps yhdistää aikaisempien menetelmien parhaat puolet ja lisää jatkuvuutta sekä automaatioastetta. DevOps edustaa kattavaa ja kehittyvää viitekehystä modernille ohjelmistokehitykselle, pyrkien parantamaan ohjelmistojen toimituksen tehokkuutta, luotettavuutta ja laatua. DevOpsin moninaisten haasteiden ymmärtäminen ja ratkaiseminen on ratkaisevan tärkeää sen onnistuneelle käyttöönotolle ja tulevalle kehitykselle. DevOpsin tavoite on tuottaa arvoa nopeammin, pienemmällä riskillä, pienemmillä muutoksilla ja paremmalla laadulla.

Asiasanat: DevOps, Ketterät menetelmät, Ohjelmistokehitys, Ohjelmistotuotanto

# ABSTRACT

Issakainen, Juuso
DevOps – History, Present and Future
Jyväskylä: University of Jyväskylä, 2024, 20 pp.
Information Systems Science, Bachelor's Thesis)
Supervisor: Kyppö, Jorma

DevOps has evolved over the past 16 years. Despite its widespread adoption, a generally accepted definition has not yet been clearly established. This literature review explores the history of software engineering, definitions of DevOps, its content, practices, and challenges. Peer-reviewed articles and studies, as well as industry literature, have been used as source material. The historical context covers the transition from the waterfall model to agile methods, highlighting the deficiencies that DevOps aims to address. The study does not find a single clear definition of DevOps. It also concludes that DevOps has evolved in response to the need to improve collaboration between software development and maintenance operations. DevOps combines the best aspects of previous methods and adds continuity and automation. DevOps represents a comprehensive and evolving framework for modern software development, aiming to improve the efficiency, reliability, and quality of software delivery. Understanding and addressing the diverse challenges of DevOps is crucial for its successful adoption and future development. The goal of DevOps is to deliver value faster, with less risk, smaller changes, and better quality.

Keywords: DevOps, Agile, Software Development, Software Engineering

# KUVAT

# SISÄLLYS

# 1 JOHDANTO

DevOps on tapa tehdä ohjelmistokehitystä. Termi DevOps on lyhennys sanoista *development* ja *operations*, jotka tarkoittavat tässä yhteydessä ohjelmistokehitystä (development) ja ylläpitoa ja/tai operointia (operations).

DevOps-termi on ollut esillä ohjelmistokehityksessä jo viimeiset 16 vuotta. Tästä huolimatta termille ei ole selkeää yksiselitteistä määritelmää. Riippuu myös asiayhteydestä ja termin käyttäjästä, että puhutaanko DevOps-toimintamallista, DevOps-menetelmistä vai DevOps-viitekehyksestä. Termistä tai sen käyttäjästä huolimatta termi DevOps on ollut pinnalla jo jonkin aikaa, ja sitä voidaan pitää tietynlaisena trendinä.

Yleisesti DevOps on muun muassa kokoelma menetelmiä, toimintoja, sääntöjä ja työkaluja, joiden avulla pyritään parantamaan jatkuvaa ohjelmistokehitystä ja digitaalisten palveluiden tuottamista. Tavoitteena on nopeampi arvon tuottaminen pienemmällä riskillä, pienemmillä muutoksilla ja paremmalla laadulla. DevOps pitää sisällään myös ajatuksen autonomisesta tiimistä, joka koostuu taidoiltaan erilaisista yksilöistä. Tämä tiimi on vastuussa tuotteen tai palvelun toteuttamisesta ideasta tuotantoon asti. Tällainen toiminta ja sen organisointi voidaan yleisesti mieltää DevOps-kulttuuriksi.

Tutkielma on toteutettu kirjallisuuskatsauksena keskittyen vertaisarvioituihin artikkeleihin ja julkaisuihin, pääpaino on uudemmilla julkaisuilla, koska ala kehittyy jatkuvasti ja uudet tutkimukset tarjoavat päivitettyä tietoa.

Tutkielma pyrkii vastaamaan seuraaviin tutkimuskysymyksiin:

1. Mitä DevOps on ja miten se määritellään?
2. Mikä on johtanut DevOpsin syntyyn?

Seuraavassa luvussa tarkennetaan käytettyä tutkimusmenetelmää ja työn vaiheita. Tutkielman kolmas pääluku käsittelee ohjelmistokehityksen historiaa ennen DevOpsia: Kuinka vesiputousmalli johti ketteriin menetelmiin. Tämän selvittäminen on tärkeää, kun neljännessä luvussa pureudutaan itse DevOpsiin ja huomataan, millä tavalla se eroaa aikaisemmista ja mitä samaa siinä on. Tässä

luvussa myös eritellään DevOpsin osa-alueet ja niiden keskeiset sisällöt. Lopuksi pohditaan johtopäätöksiä, heränneitä kysymyksiä ja tiivistetään havainnot yhteenvedoksi.

# 2 TUTKIMUSMENETELMÄ

Tämä tutkielma on suoritettu kirjallisuuskatsauksena. Aineiston hakemiseen on käytetty JYKDOK-järjestelmää ja tämän kautta myös Google Scholaria. Hakusanoja ovat olleet:

- ”DevOps”
- ”History of software development”
- ”DevOps in practice”
- ”Agile”

Lisäksi löydetyissä aineistoissa esiintyneitä lähteitä on hyödynnetty.

Aineiston valinnassa on käytetty merkityksen ja sisällön sopivuutta tutkimuskysymyksiin tiivistelmän ja yleiskuvan perusteella. Uudempia tutkimuksia on suosittu relevantin tutkimustiedon takia. Joissain kappaleissa on käytetty tutkielman tekijän yleistä tietämystä ja etsitty asiaankuuluva lähde yllä mainittujen hakusanojen ulkopuolelta.

Itse tutkimustyö voidaan jakaa neljään vaiheeseen: suunnittelu, aineiston kerääminen, kirjoitustyö ja viimeistely. Suunnitteluvaiheessa perehdyttiin aikaisempiin töihin ja tutkimuksiin sekä kirjoitettiin johdanto. Tältä pohjalta todettiin, että DevOps on edelleen määritelmältään monitulkintainen ja myös hyvin kiinnostava aihe. Seuraavaksi valittiin yllä mainitut hakusanat ja etsittiin työhön sopivia artikkeleita, julkaisuja ja kirjoja. Kirjoitusvaiheessa työn sisällysluettelo muotoutui karkeasti DevOpsin määritelmän ympärille. Historia ja käytäntö antoivat kontekstin ja sopivan käytännöllisyyden. Tämä menetelmäkappale kirjoitettiin vasta viimeistelyvaiheessa, jolloin myös johdanto ja yhteenveto valmistuivat. Palautetta tähän työhön saatiin tutkielman tekijän läheisiltä sekä työn ohjaajalta. Palautteen avulla tämä työ muodostui selkeämmäksi ja jäsennellymmäksi kokonaisuudeksi, jota on miellyttävää lukea. Viimeistelyvaiheessa joitain lähteitä tarkasteltiin uudestaan ja etsittiin uusia näkökulmia tarkemmilla hauilla alussa mainituista tietolähteistä sekä päivitettiin johdanto ja tiivistelmä.

# 3 OHJELMISTOTUOTANNON HISTORIA

Ohjelmistotekniikka on tietotekniikan tieteenhaara, joka tutkii ohjelmistojen kehittämistä ja siihen liittyviä kysymyksiä. Sen osa-alueet ovat ohjelmistotuotanto (Software Engineering) ja tietojenkäsittelytiede (Computer Science). Tässä työssä keskitytään ohjelmistotuotantoon, joka on yhteisnimitys menetelmille, joita käytetään tietokoneohjelmien ja -ohjelmistojen tuottamiseen. Ohjelmistotuotannon historia yltää 1950-luvulta tähän päivään. Tässä luvussa käymme läpi tätä historiaa, sen päätapahtumia ja lisäksi peruskäsitteitä.

Ohjelmistotuotanto tieteenalana sai alkunsa, kun ohjelmistoprojektien monimutkaisuus ja laajuus alkoivat kasvaa nopeasti 1900-luvun puolivälissä. Tieteellinen lähestymistapa oli välttämätön, ja tätä tarvetta korostettiin erityisesti vuoden 1968 NATO:n ohjelmistotuotannon konferenssissa, jossa käsiteltiin ”ohjelmistokriisiä”. Tämä kriisi kuvasi perinteisten projektinhallintatekniikoiden kyvyttömyyttä käsitellä ohjelmistoprojektien monimutkaisuutta. (Naur & Randell, 1969)

1970-luvulla ohjelmistotuotannon metodologiat alkoivat muotoutua. Winston W. Royce esitteli kuuluisan vesiputousmallin vuonna 1970, mikä oli ensimmäinen systemaattinen, sekventiaalinen lähestymistapa ohjelmistotuotantoon. Tällöin termiä vesiputousmalli ei tosin vielä käytetty. (Royce, 2021) 1980-luvulla nähtiin ketterien menetelmien varhaisia ilmentymiä, kuten Scrum ja Extreme Programming (XP), jotka olivat nopeampia ja joustavampia kuin perinteiset kehitysmenetelmät (Schwaber & Beedle, 2001).

Vuonna 2001 julkaistu Agile Manifesto (*Manifesto for Agile Software Development*, 2001) oli merkittävä käännekohta ohjelmistotuotannon historiassa. Tämä manifesti määritteli ketterän ohjelmistotuotannon perusperiaatteet, kuten prosessien joustavuuden, tiiviin asiakasyhteistyön ja kyvyn reagoida muutoksiin nopeasti. Ketterät menetelmät, kuten Scrum, Lean ja XP, tulivat standardiksi monille kehitystiimeille ympäri maailmaa. (Highsmith, 2002)

Viime vuosina ohjelmistotuotanto on jatkanut evoluutiotaan kohti entistä integroidumpia käytäntöjä, kuten DevOps (Kim ym., 2021). Tekoälyn ja koneoppimisen soveltaminen ohjelmistotuotannossa avaa myös uusia mahdollisuuksia

esimerkiksi automatisoinnissa ja tehokkuudessa (Li ym., 2018). Seuraavassa luvussa syvennytään DevOpsiin.

# 4 DEVOPS

DevOps on tapa tehdä ohjelmistokehitystä. DevOps on määritelty useilla tavoilla ja eri painotuksilla. Samoin sisältöä on kuvattu eri sanoin ja kuvauksin. Tässä luvussa käydään läpi määritelmiä, sisältöä, käytäntöä ja haasteita.

## 4.1 Määritelmä

DevOps-termin käyttö ja tutkimus alkoi, kun Patrick Debois esitteli sen konferenssissa: ”Agile Infrastructure and Operations” vuonna 2008 (Lwakatare, 2017). DevOps pyrkii yhdistämään kehitys- ja ylläpitotiimit (Leite ym., 2020). ”DevOps on organisaation lähestymistapa, joka korostaa empatiaa ja ristiin toiminnallista yhteistyötä tiimeissä ja niiden välillä, erityisesti kehitys- ja ylläpitotiimien ohjelmistokehitysorganisaatioissa, jotta voidaan operoida joustavia järjestelmiä ja nopeuttaa muutosten julkaisua.” (Dyck ym., 2015, oma suomennos). ”DevOps on yhteistyöllinen ja monitieteellinen pyrkimys organisaatiossa automatisoida jatkuvasti uusien ohjelmistoversioiden toimitusta, samalla taaten niiden oikeellisuuden ja luotettavuuden.” (Leite ym., 2020, oma suomennos). DevOpsin ytimessä on kommunikaation, yhteistyön ja synergian parantaminen IT-tiimeissä, jotta sovelluksia voidaan kehittää ja parantaa vastaamaan markkinan ja liiketoiminnan muuttuviin vaatimuksiin (Lwakatare ym., 2019). Lisäksi ”DevOps on globaalisti hyväksytty kulttuuri ja kokoelma käytänteitä. DevOps-kulttuuri keskittyy lean-periaatteisiin, automaatioon, mittaukseen ja tiedon jakamiseen. Ihmisillä on merkittävä rooli kulttuurin kehittämisessä ja tiedon jakamisessa.” (Faustino ym., 2022, oma suomennos). Määritelmästä riippuen korostetaan joko yhteistoimintaa, automaatiota tai vastaamista muuttuvaan ympäristöön. Tämä osoittaa, että DevOps on laaja kokonaisuus ja sen käyttö on usein tilanteen tai kontekstin sanelemaa, jolloin eri painotukset ovat tarpeen.

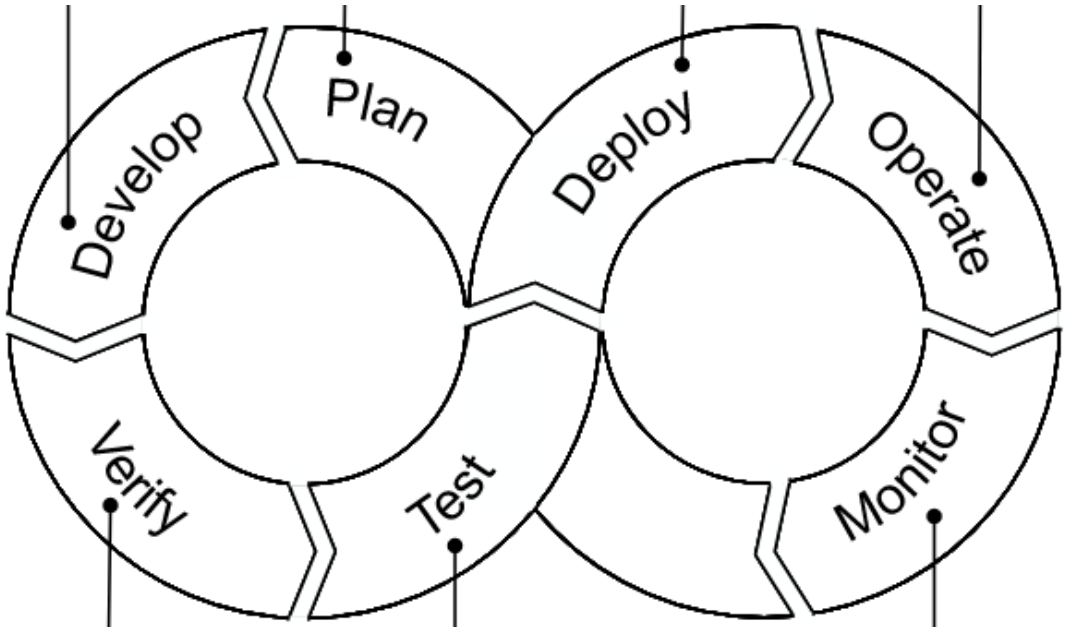
Määritelmien moninaisuudesta huolimatta DevOpsin tavoitteena on saavuttaa liiketoiminnallisia tuloksia, kuten riskin ja kustannusten vähentäminen, säännösten ja lakien noudattaminen, tuotteiden laadun parantaminen ja

asiakastyytyväisyyden lisääminen. Keino näiden tavoitteiden saavuttamiseksi on prosessi, joka sisältää usein ja luotettavasti toteutettuja julkaisuja. Jatkuva toimitus ja jatkuva julkaisu johtavat parempaan laatuun ja asiakastyytyväisyyteen lyhyen palautesyklin ansiosta – pieniin nopeisiin muutoksiin saadaan nopeasti palautetta monitoroimalla ja testaamalla. (Leite ym., 2020)

Pieniä muutoksia, mahdollisimman usein, yhteistyötä korostaen, automatisoidusti ja luotettavasti. Verrattuna vesiputousmalliin, jossa jokainen vaihe suoritetaan loppuun ennen seuraavan alkamista, tämä lähestymistapa on hyvin erilainen. Agilesta tuttu yhteistyö ja muutoksiin vastaaminen on selkeästi sisällytetty DevOpsiin, samoin kuin ihmisten ja vuorovaikutuksen tunnistaminen ja korostaminen. Niin on myös vesiputousmallissa esiintyneet vaiheet, joista lisää seuraavassa luvussa.

## 4.2 Sisältö

DevOps kuvataan usein päättymättömänä syklinä, jossa eri vaiheet seuraavat toisiaan ja toistuvat. Nämä vaiheet ovat suunnittelu (plan), kehitys (development), tarkistus (verify), testaus (test), julkaisu (deploy), operointi (operate) ja monitorointi (monitor). Nämä vaiheet ovat hyvin samankaltaisia kuin perinteisessä ohjelmistotuotannossa – erona jatkuvuus ja korkea automaatioaste. (Alnafessah ym., 2021)




**Kuva 1 –** DevOpsin vaiheet (Alnafessah ym., 2021)

Suunnitteluvaihe pyrkii määrittämään tavoitteet ja vaatimukset ohjelmistolle ja se tuotannolle. Tässä vaiheessa tehdään myös alustava suunnitelma päivityksille ja julkaisuille eri iteraatioissa. Kehitysvaiheessa ohjelmakoodi luodaan ja arvioidaan iteratiivisesti keskittyen myös infrastruktuuriin. Tyypillisesti

ohjelmakoodi luodaan pienissä osissa, jolloin muutoksia voidaan testata yksikkö- ja integraatiotasolla jatkuvasti. Tarkistusvaiheessa ohjelmiston osien oikeellisuus suhteessa vaatimuksiin tarkastetaan. Testaus sisältää automaattisia testejä, joita ajetaan ja päivitetään jatkuvasti. Tavoitteena on varmistua ohjelmiston laadusta. Tässä vaiheessa voidaan myös julkaista ominaisuuksia eri käyttäjäryhmille palautteen saamiseksi. Julkaisu on jatkuvaa ja usein automaattista. Testatut ohjelmistomuutokset viedään tuotantoympäristöön käyttäjien saataville. Operoinnissa keskitytään julkaisun jälkeiseen sovelluksen konfiguraatioiden hallintaan. Näitä ovat muun muassa resurssien hallintaan ja automaattinen skaalautuminen. Monitoroinnissa kerätään dataa tuotannosta. Datan avulla pyritään arvioimaan sovelluksen suorituskykyä ja löytämään poikkeuksia. Tämä palaute auttaa kehittämään sovellusta. (Alnafessah ym., 2021)

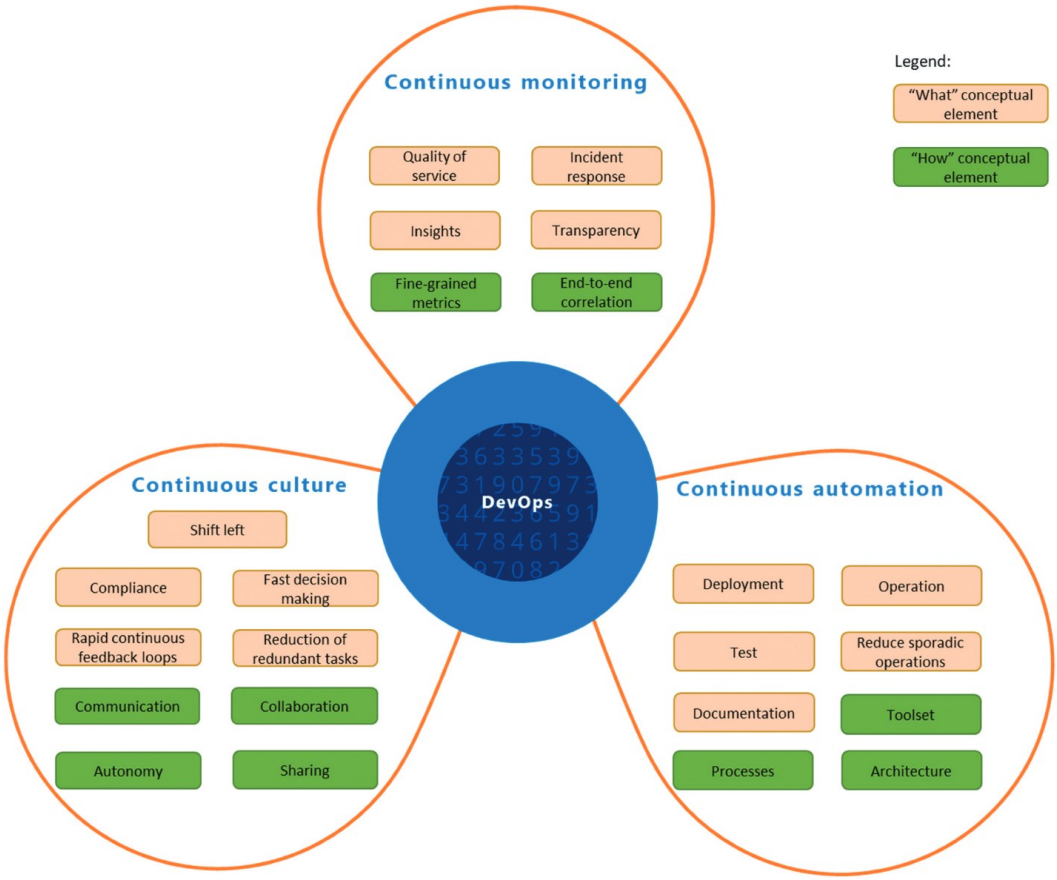
Alkuperäisessä Roycen (Royce, 2021) vesiputousmallissa 1970-luvulta vaiheet ohjelmistokehityksen vaiheet olivat:

1. Vaatimusten määrittely
2. Suunnittelu
3. Toteutus
4. Integraatio
5. Testaus
6. Asennus
7. Ylläpito

Tämä malli on ennen kaikkea vaiheellinen prosessi, jossa edellinen vaihe valmistui ennen seuraavan alkamista. DevOpsissa korostetaan automaatiota ja jatkuvuutta, sekä kehitetään inkrementaalisesti ja iteratiivisesti pienissä osissa, aina varmistuen muutosten oikeellisuudesta ja tuotteen toimivuudesta käyttötarkoitukseensa. DevOpsissa ei myöskään ole selkeää vaihetta vaatimusten määrittelylle, sen voidaan olettaa kuuluvan suunnitteluun ja tarkentuvan tai muuttuvan jokaisella kierroksella syklissä. Näin toimien saatu ja kerätty palaute voidaan nopeasti ja ketterästi sisällyttää tuleviin julkaisuihin. Seuraavassa luvussa perehdytään näiden vaiheiden ja niiden jatkuvuuden käyttöönottoa.

## 4.3 Käytäntö

DevOpsin käyttöönotto keskittyy kehitys- ja ylläpitotiimien välisen kuilun häivyttämiseen, joka edistää yhteistyötä ja nopeaa ohjelmistotuotantoa ilman laadun ja ylläpidon heikkenemistä. Yleisimpiä hyötyjä ovat nopeampi pääsy markkinoille, synergian parantuminen, automaation parantuminen, vähentyneet epäonnistumiset ja vähentyneet tietoturvaongelmat. (Faustino ym., 2022)




**Kuva 2** - DevOps osa-alueet (Gall & Pigni, 2022)

Michael Gall ja Federico Pigni jakavat DevOpsin kolmeen osa-alueeseen artikkelissa ”Taking DevOps Mainstream: A Critical Review and Conceptual Framework” (Gall & Pigni, 2022). Tämän konseptuaalisen viitekehyksen osat ovat jatkuva kulttuuri, jatkuva automaatio ja jatkuva monitorointi. Keskittymällä näihin osa-alueisiin ja niihin liittyviin käytäntöihin, organisaatiot voivat toteuttaa DevOpsia tehokkaammin. Seuraavissa kappaleissa määritellään nämä osa-alueet, niiden tavoitteet ja miten ne saavutetaan.

Jatkuva kulttuuri tarkoittaa DevOps-tiimin jaettua visiota ja käyttäytymistä, joka korostaa yhteistyötä, viestintää ja sitoutumista jatkuvaan palautteeseen. Tavoitteena on edistää yhteistyötä ja jaettua vastuuta, sekä mahdollistaa nopea päätöksenteko ja sopeutumiskyky. Keinoja näihin tavoitteisiin ovat jatkuvat palautesilmukat, autonomian ja yhteistyön edistäminen sekä jatkuvan parantamisen ja oppimisen kulttuurin ylläpito. Kulttuuria ei tule nähdä yhtenä elementtinä, vaan kategoriana, joka sisältää erilaisia käytäntöjä, jotka tukevat jatkuvaa parantamista. (Gall & Pigni, 2022)

Työkalujen ja prosessien käyttö toistuvien tehtävien automatisoimiseksi koko ohjelmistokehityksen elinkaaren ajan on jatkuvaa automaatiota. Tavoite on vähentää manuaalista työtä ja lisätä julkaisujen tiheyttä ja luotettavuutta. Nämä tavoitteet saavutetaan ottamalla käyttöön automaatiotyökaluja julkaisuun,

testaukseen ja monitorointiin. Tehokkaan automaation mahdollistaa mikropalveluarkkitehtuuri. (Gall & Pigni, 2022)

Seuraamalla järjestelmän suorituskykyä, anomalioita ja virheitä monitorointityökalujen avulla voidaan havaita ongelmat ennen kuin ne vaikuttavat loppukäyttäjiin. Arvioimalla järjestelmän laatua ja tiimin suorituskykyä hienojakoisilla mittareilla sekä luomalla prosessit jatkuvaan arviointiin ja iteratiiviseen parantamiseen voidaan toteuttaa jatkuvaa monitorointia. (Gall & Pigni, 2022)

Arvioidessaan DevOpsin eri vaiheiden merkitystä Ankur Kumar, Mohammad Nadeem ja Mohammad Shameem tunnistivat kuusi eri ydinvaihetta ja tutki näiden suhteellista merkitystä. Nämä vaiheet ovat: jatkuva suunnittelu, jatkuva kehitys, jatkuva testaus, jatkuva integrointi, jatkuva julkaisu ja jatkuva monitorointi. Näistä vaikuttavimmaksi osoittautui jatkuva integrointi, mutta kirjoittajat toteavat, että DevOps tulee nähdä kokonaisvaltaisena prosessina, jossa jokainen vaihe tukee muita vaiheita. (Kumar ym., 2024) Jatkuvuuden lisääminen eri vaiheisiin lisää tarvetta automaatiolle ja yhteistyölle sekä tuo mukanaan haasteita, joista lisää seuraavaksi.

## 4.4 Haasteet

Konsepteja ja haasteita voidaan käsitellä esimerkiksi kolmesta eri näkökulmasta, kuten selvityksessä ”A Survey of DevOps Concepts and Challenges” (Leite ym., 2020) on tehty. Nämä kolme näkökulmaa ovat tekninen, organisatorinen ja käytännöllinen. Niiden alle asettuvat eri konseptit ja näkökulmat seuraavasti.

Tekniset näkökohdat:

- Käytettävät automaatiotyökalut ovat keskeisiä, ja niiden ymmärtäminen on tärkeää toteutettaessa DevOps-menetelmiä mahdollisimman tehokkaasti.
- Mikropalveluarkkitehtuurit ja sovellusten eristäminen ovat tärkeitä elementtejä, ja ne mahdollistavat joustavamman ja nopeamman ohjelmistokehityksen ja -julkaisun.
- Haasteita tuovat muun muassa erilaisten työkalujen ja tekniikoiden yhdistäminen ja hallinta.

Organisatoriset haasteet:

- DevOps vaatii kulttuurin muutosta organisaatiossa, jossa kehitys- ja ylläpitotiimit työskentelevät yhdessä.
- Organisaation rakenteen muuttaminen DevOpsin mukaiseksi on haastavaa, ja siihen liittyy erilaisia strategioita, kuten tiimien välisen yhteistyön tehostaminen ja uusien tiimien muodostaminen.
- Jatkuva koulutus ja osaamisen kehittäminen ovat välttämättömiä.

Käytännön toteutukset:

- DevOpsia sovelletaan monin eri tavoin, ja sen toteutus vaihtelee yritysten ja projektien mukaan.
- Onnistuneen muutoksen avain on ymmärtää sen tuomat hyödyt, kuten nopeampi julkaisu ja parantunut ohjelmiston laatu, sekä kohdata sen tuomat haasteet, kuten tarve jatkuvaan oppimiseen ja kulttuurin muutokseen.

DevOps on moniulotteinen ja jatkuvasti kehittyvä alue, joka vaatii jatkuvaa tarkastelua ja sopeutumista muuttuviin teknologisiin, liiketoiminnallisiin ja organisatorisiin tarpeisiin. Moninaisten haasteiden ja mahdollisuuksien ymmärtäminen on keskeistä muutoksen onnistumiselle. (Leite ym., 2020) Voidaan olettaa, ettei muutos tule koskaan olemaan lopullinen. Globaali markkina, muuttuvat teknologiat ja tarpeet vaativat jatkuvaa muutosta.

# 5 POHDINTAA JA YHTEENVETO

## 5.1 Pohdintaa

Olen itse työskennellyt ohjelmistotuotannon parissa yli 10 vuotta. DevOps-termin kuulin ensimmäistä kertaa vuonna 2013. Sen määritelmä ja sovellutukset olivat silloin epäselviä ja ovat edelleen. Tavoitteet laajasti katsoen ovat selviä - tehdä nopeammin ja laadukkaammin. Usein tosielämässä nämä asiat ovat ristiriidassa. Siinä mielessä DevOps on ja tulee olemaan mielenkiintoinen viitekehys, jolla tätä jatkuvasti kehittyvää ja kasvavaa toimialaa yritetään määrittää.

Tutkimusta tehdessä minulle valkeni, ettei ole olemassa mitään yhtä DevOpsia. On erilaisia määritelmiä, eri painotuksin. On myös erilaisia sisällöllisiä määritelmiä ja kategorisointeja. Näitä on menestyksekkäästi tutkittu ja löydetty joitain yhteisiä nimittäjiä ja toistuvia käsitteistöjä. Mielenkiintoista olisi nähdä tutkimus, jossa kerrotaan mitä DevOps ei ole. Tulevaisuudessa voisi olla helpompaa tutkia ja hyödyntää DevOpsia, jos se voitaisiin määritellä standardiksi tai määritellä selkeästi niin, että kaikki jakaisivat saman määritelmän.

Oma määritelmäni on: ”DevOps on kaikki mitä tarvitaan, jotta idea saadaan tuotantoon tehokkaasti, laadukkaasti ja toistettavasti.” Organisaatioiden tulisi nähdä DevOps ajatusmallina, jossa jatkuvuus ja muutos ovat keskiössä.

## 5.2 Yhteenveto

DevOps on kehittynyt vastauksena tarpeeseen parantaa ohjelmistokehityksen ja ylläpitotoimintojen välistä yhteistyötä, nopeutta ja luotettavuutta. Historia osoittaa, että ohjelmistotuotannon monimutkaisuuden kasvu johti tieteelliseen lähestymistapaan ja menetelmien kehittämiseen, kuten vesiputousmalli ja ketterät menetelmät. DevOps yhdistää näiden menetelmien parhaat puolet ja keskittyy jatkuvaan parantamiseen korkean automaatioasteen ja yhteistyön avulla.

DevOpsilla ei ole selkeää yksiselitteistä määritelmää. Se määritellään yleisesti tavaksi tehdä ohjelmistokehitystä tehokkaasti ja laadukkaasti.

DevOpsin käyttöönotto vaatii merkittäviä kulttuurisia ja teknologisia muutoksia organisaatioissa. Onnistuneen DevOps-muutoksen avain on ymmärtää sen tuomat hyödyt, kuten nopeampi julkaisu ja parantunut ohjelmiston laatu, sekä kohdata sen tuomat haasteet, kuten tarve jatkuvaan oppimiseen ja kulttuurin muutokseen. DevOps on moniulotteinen ja jatkuvasti kehittyvä alue, joka vaatii jatkuvaa tarkastelua ja sopeutumista muuttuviin teknologisiin, liiketoiminnallisiin ja organisatorisiin tarpeisiin. Tulevaisuudessa DevOpsin rooli ohjelmistokehityksessä tulee todennäköisesti kasvamaan entisestään, mikä tekee siitä keskeisen osan modernia ohjelmistotuotantoa. On myös mahdollista, että DevOpsin käytäntöjä ja periaatteita voidaan soveltaa muillakin aloilla.

# LÄHTEET

Alnafessah, A., Gias, A. U., Wang, R., Zhu, L., Casale, G., & Filieri, A. (2021). Quality-Aware DevOps Research: Where Do We Stand? IEEE Access, 9, 44476–44489. https://doi.org/10.1109/ACCESS.2021.3064867

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

Faustino, J., Adriano, D., Amaro, R., Pereira, R., & da Silva, M. M. (2022). DevOps benefits: A systematic literature review. Software: Practice and Experience, 52(9), 1905–1926. https://doi.org/10.1002/spe.3096

Gall, M., & Pigni, F. (2022). Taking DevOps mainstream: A critical review and conceptual framework. European Journal of Information Systems, 31(5), 548–567. https://doi.org/10.1080/0960085X.2021.1997100

Highsmith, J. A. (2002). Agile Software Development Ecosystems. Addison-Wesley Professional.

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution.

Kumar, A., Nadeem, M., & Shameem, M. (2024). Assessment of DevOps Lifecycle Phases and their Role in DevOps Implementation using Best–Worst MCDM. International Journal of Information Technology, 16(4), 2139–2147. https://doi.org/10.1007/s41870-023-01566-3

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2020). A Survey of DevOps Concepts and Challenges. ACM Computing Surveys, 52(6), 1–35. https://doi.org/10.1145/3359981

Li, X., Jiang, H., Ren, Z., Li, G., & Zhang, J. (2018). Deep Learning in Software Engineering (arXiv:1805.04825). arXiv. https://doi.org/10.48550/arXiv.1805.04825

Lwakatare, L. E. (2017). DevOps adoption and implementation in software development practice: Concept, practices, benefits and challenges. University of Oulu.

Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., Kuvaja, P., Mikkonen, T., Oivo, M., & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. Information and Software Technology, 114, 217–230. https://doi.org/10.1016/j.infsof.2019.06.010

Manifesto for Agile Software Development. (2001). https://agilemanifesto.org/

Naur, P., & Randell, B. (1969). Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO.

Royce, W. W. (2021). Managing the Development of Large Software Systems (1970). https://doi.org/10.7551/mitpress/12274.003.0035

Schwaber, K., & Beedle, M. (2001, marraskuuta 10). Agile Software Development with Scrum (world). Guide Books. https://doi.org/10.5555/559553