



Oskari Närhi

# Azure DevOps keskikokoisessa yrityksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

13.11.2024

# Tiivistelmä

Tekijä: Oskari Närhi  
Otsikko: Azure DevOps keskikokoisessa yrityksessä  
Sivumäärä: 28 sivua  
Aika: 13.11.2024

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintätekniikka  
Ammatillinen pääaine: Ohjelmistotuotanto  
Ohjaajat: FM Simo Silander  
Tekninen johtaja, Pepe Salo

---

Tässä Insinööriyössä tarkastellaan Azure DevOps -ympäristön käyttöönottoa keskikokoisessa suomalaisessa yrityksessä, Kastenilla, jonka toiminta keskittyy logistiikka- ja varastoratkaisuihin. Työn tarkoituksena on selvittää, miten Azure DevOps voi tukea ohjelmistokehityksen työkujuja, parantaa yhteistyötä ja tehostaa dokumentaation hallintaa organisaatiossa, jossa on noin 50 työntekijää.

Insinööriyö tarjoaa yleiskatsauksen DevOpsin ja erityisesti Azure DevOpsin keskeisiin käsitteisiin, kuten CI/CD (jatkuva integraatio ja jatkuva toimitus), tehtävienhallintaan ja dokumentaation keskittämiseen. Kastenilla käyttöönotto alkoi tarpeesta yhdistää eri työkalut yhteen keskitettyyn järjestelmään, sillä aiemmin dokumentaatio ja projektinhallinta oli hajautettu useisiin alustoihin, mikä vaikeutti tiedon hallintaa ja jakamista.

Työn analyysi osoittaa, että Azure DevOpsin tarjoamat työkalut, kuten Azure Boards, Repos ja Pipelines, ovat parantaneet tehtävien hallintaa, työn seurattavuutta ja yhteistyön läpinäkyvyyttä. Lisäksi CI/CD-putken suunnittelu ja automaatiomahdollisuudet tukevat laadunvarmistusta ja nopeuttavat ohjelmistojen tuotantoon vientiä. Käyttöönottoprosessi on kuitenkin tuonut mukanaan myös haasteita, kuten oppimiskäyrän ja muutosvastarinnan, joita on pyritty ratkaisemaan koulutuksella ja jatkuvalla kehityksellä.

Insinööriyö päättyy johtopäätöksiin, joissa korostetaan Azure DevOpsin soveltuvuutta keskikokoisille yrityksille sekä suosituksiin muille organisaatioille, jotka harkitsevat DevOpsin käyttöönottoa. Näihin suosituksiin kuuluvat muun muassa käyttöönoton vaiheistus, koulutuksen merkitys ja palautteen hyödyntäminen järjestelmän kehittämisessä.

Avainsanat: DevOps, Azure, Tehtävien hallinta

## Abstract

Author: Oskari Närhi  
Title: Azure DevOps in Medium-Sized Company  
Number of Pages: 28 pages  
Date: 13 November 2024

Degree: Bachelor of Engineering  
Degree Programme: Information and Communication Technology  
Professional Major: Software Engineering  
Supervisors: Simo Silander, M.Sc.  
Technical Director, Pepe Salo

---

This thesis examines the implementation of the Azure DevOps environment in a medium-sized Finnish company, Kasten, which specializes in logistics and warehouse solutions. The purpose of the study was to determine how Azure DevOps can support software development workflows, improve collaboration, and enhance documentation management in an organization with approximately 50 employees.

The thesis provides an overview of the key DevOps concepts, particularly those related to Azure DevOps, such as CI/CD (continuous integration and continuous delivery), task management, and centralized documentation. The implementation at Kasten began with a need to consolidate various tools into a centralized system, as documentation and project management were previously scattered across multiple platforms, making information management and sharing difficult.

The analysis shows that the tools offered by Azure DevOps, such as Azure Boards, Repos, and Pipelines, improved task management, work tracking, and collaboration transparency. Additionally, the design of the CI/CD pipeline and automation capabilities support quality assurance and accelerate the deployment of software to production. However, the implementation process presented also challenges, such as the learning curve and resistance to change, which were addressed through training and continuous development.

The thesis concludes with insights that emphasize the suitability of Azure DevOps for medium-sized companies, as well as recommendations for other organizations considering DevOps implementation. The recommendations include phased implementation, importance of training, and leveraging feedback to develop the system further.

Keywords: DevOps, Azure, Task management

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Kastenin toimintaympäristö ja DevOpsin käyttöönotto	2
2.1	Historia	2
2.2	Ohjelmistokehitys	3
2.3	DevOps-perusteet ja Azure DevOpsin komponentit	4
2.3.1	DevOpsin perusteet	5
2.3.2	Azure DevOps -komponentit	6
2.4	Käyttöönoton tavoitteet	7
3	Tehtävien hallinta ja yhteistyön tehostaminen	8
3.1	Azure Boards ja työnkulun seuranta	9
3.1.1	Tehtävien hallinta ja priorisointi	9
3.1.2	Käyttäjätasojen hallinta	10
3.1.3	Tiimipalavereiden hyödyntäminen	11
3.2	Hiljaisen tiedon näkyväksi tekeminen ja dokumentaation keskittäminen	11
3.3	Digitaalinen muistijälki	12
4	Prosessien optimointi ja laadunvarmistus	12
4.1	Queryt ja dashboardit	12
4.2	Ulkopuoliset työkalut	14
4.2.1	Power Automate	15
4.2.2	PowerShell	17
4.3	Laadunvarmistuksen nykyiset käytännöt	17
4.4	CI/CD-putkien suunnittelu ja automatisointi	19
5	Lopputulokset, haasteet ja suositukset	20
5.1	Azure DevOpsin tuomat hyödyt Kastenille	20
5.2	Käyttöönottoprosessin haasteet ja ratkaisut	21
5.3	Käyttäjäpalaute ohjelmistokehityksen ulkopuolella	22
5.4	Lopputulokset ja opit	22
5.5	Suosituksset keskikokoisille yrityksille	23

6 Yhteenveto

24

Lähteet

26

## Lyhenteet

- CI: *Continuous Integration*. Koodin yhtenäistäminen ja automaattisen testauksen järjestäminen.
- CD: *Continuous Delivery*. Jatkuva toimitus. Koodi on aina valmiina julkaistavaksi tuotantoon.
- DevOps: *Development and Operations*. Toimintamalli, joka yhdistää kehitys- ja operatiiviset toiminnot yhdeksi yhteistyötä ja automatisointia painottavaksi prosessiksi.
- JSON: *JavaScript Object Notation*. Yksinkertainen ja kevyt avoimen standardin tiedostomuoto tiedonvälitykseen ja tallennukseen.
- PAT: *Personal Access Token*. Käyttäjakohtainen tunniste, joka mahdollistaa ulkoisille työkaluille turvallisen pääsyn Azure DevOps -resursseihin ilman käyttäjätunnusta ja salasanaa.
- PLC: *Programmable Logic Controller*. Ohjain, jota käytetään usein teollisuuden automaatiojärjestelmissä.
- PR: *Pull request*. Prosessi, jossa kehittäjä ehdottaa koodimuutosten yhdistämistä päähaaraan tai muuhun haaraan tarkistuksen ja hyväksynnän jälkeen.
- Päähaara: *Master branch*. Ohjelmistokehityksen pääasiallinen koodiversio, johon hyväksytyt muutokset yhdistetään ja josta tuotantoversiot yleensä julkaistaan.

# 1 Johdanto

Ohjelmistokehityksessä on viime vuosina tapahtunut merkittävä muutos kohti nopeampia ja ketterämpiä toimintatapoja. Yksi keskeisimmistä trendeistä on DevOps, toimintamalli, jonka tarkoituksena on yhdistää kehitys- ja operatiiviset tiimit ja mahdollistaa jatkuva, tehokas ohjelmiston tuotantoprosessi [1].

DevOpsin yleistymisen taustalla on pyrkimys lyhentää ohjelmistojen toimitusaikoja, parantaa niiden laatua ja helpottaa IT-infrastruktuurin hallintaa. [2.]

Tässä työssä tarkastellaan Azure DevOps -ympäristön käyttöönottoa keskikokoisessa yrityksessä ja analysoidaan, kuinka tämä ympäristö voi parantaa tiedonhallintaa, yhteistyötä ja työn seurattavuutta ohjelmistokehityksessä. Työ keskittyy Azure DevOpsin keskeisiin ominaisuuksiin sekä sen vaikutuksiin kohdeyrityksen työnkulkuun ja tiedonhallintaan. Lisäksi työ tarjoaa suosituksia muille keskikokoisille yrityksille, jotka harkitsevat DevOpsin käyttöönottoa ohjelmistokehitysprosessinsa tehostamiseksi.

Työn päätavoitteena on selvittää, miten Azure DevOps soveltuu keskikokoisen, noin 50 henkilön yrityksen tarpeisiin. Azure DevOpsia kuitenkin käyttää vain puolet yrityksen henkilöstöstä, joista valtaosa on muita kuin ohjelmistokehittäjiä. Työ analysoi käyttöönoton hyötyjä ja haasteita sekä tarjoaa käytännön esimerkkejä siitä, kuinka Azure DevOps on tukenut ohjelmistokehityksen eri vaiheita kohdeyrityksessä Kastenilla. Erityisesti työ keskittyy siihen, miten DevOps-ympäristön käyttö mahdollistaa eri vastuualueita omaavien tiimien välisen yhteistyön, dokumentaation keskittämisen ja tehtävien tehokkaamman seurannan.

Kohdeyritys Constructor Finland Oy - Kasten on suomalainen metallialan yritys, joka toimii Lohjalla ja keskittyy logistiikka- ja varastoratkaisuihin. Yrityksen brändinimenä on Kasten, johon työssä viitataan jatkossa. Kastenin liikevaihdosta osa muodostuu perinteisistä hyllyratkaisuista ja osa koneista,

joista merkittävimmät ovat varastoautomaatteja. [3.] Azure DevOpsin käyttäjäkunta yrityksessä koostuu ainoastaan koneiden parissa työskentelevästä henkilöstöstä.

Yrityksessä oli ennen Azure DevOpsin käyttöönottoa käytössä hajautettu dokumentaatio- ja tehtävienhallintajärjestelmä, jossa tieto oli jaettu eri työkaluihin, kuten Exceleihin, SharePointeihin, DropBoxeihin ja sisäisiin verkkolevyihin. Tämä aiheutti haasteita tiedon hallinnassa ja yhteistyössä, ja DevOpsin käyttöönoton tavoitteena oli keskittää dokumentaatio ja tehtävien seuranta yhteen paikkaan, parantaa tiedon saavutettavuutta ja vähentää manuaalista työtä.

## **2 Kastenin toimintaympäristö ja DevOpsin käyttöönotto**

Kastenin pitkän historian ja ohjelmistokehityksen kehityskaaren ymmärtäminen auttaa hahmottamaan niitä lähtökohtia, joiden pohjalta yritys otti käyttöön Azure DevOps -ympäristön. Yrityksen historia, varastointiratkaisujen pitkä perinne ja viime vuosikymmenten teknologiset muutokset ovat muokanneet Kastenin toimintatapoja ja tuotekehityksen prosesseja. Lisäksi tässä luvussa tarkastellaan DevOps-toimintamallin perusteita sekä Azure DevOps -ympäristön käyttöönoton keskeisiä tavoitteita Kastenilla.

### **2.1 Historia**

Kastenin juuret ulottuvat yli 130 vuoden päähän vuoteen 1886, jolloin Hjalmar Henrik Kasten perusti läkkisepänliikkeen Helsinkiin. Yritys aloitti toiminnan metallialalla ja kasvoi vuosikymmenten saatossa laajemmaksi teolliseksi toimijaksi. Vuonna 1951 Kasten keskittyi ensimmäistä kertaa varastointiratkaisujen valmistukseen, ja siitä lähtien varastointi- ja logistiikkaratkaisut ovat olleet yrityksen keskeisiä tuotteita. [4.]

Kasten on kokenut useita omistajavaihdoksia historiansa aikana, mutta tuotemerkki ja brändinimi ovat säilyneet samana. Tämä jatkuvuus on auttanut

yrittäjästä säilyttämään vahvan aseman varastointiratkaisujen markkinoilla, ja Kastenin brändi on edelleen tunnettu laadukkaista ja luotettavista tuotteistaan. Vuonna 2003 Kasten toi markkinoille ensimmäiset varastoautomaattinsa, jotka olivat merkittävä askel yrityksen tuotevalikoiman ja teknologisen osaamisen kehityksessä. Varastoautomaatit yhdistivät automaation ja varastoinnin, ja ne edustavat yrityksen modernimpaa tuotantolinjaa ja tuotekehitystä. [4.]

## 2.2 Ohjelmistokehitys

Vaikka Kastenilla on pitkä historia varastointiratkaisujen kehittämisessä, ohjelmistokehityksen osalta yrityksen prosessit ovat olleet haasteellisia ja monin paikoin epäyhtenäisiä. Perinteisesti ohjelmistokehitys ei ole ollut kovin järjestelmällistä, ja mitään vakiintuneita prosesseja ei ollut käytössä. Tämä toi haasteita erityisesti silloin, kun ohjelmistot alkoivat integroitua yhä enemmän varastointijärjestelmiin ja varastoautomaatteihin. [5; 6.]

Aiemmin yrityksen ohjelmistokehityksessä käytetyt versiohallintaratkaisut olivat yksinkertaisia ja paikallisia. Koodirepositoriot sijaitsivat paikallisella palvelimella, ja versiohallinta oli manuaalista, perustuen kehittäjien huolellisuuteen ja muistinvaraiseen muutosten seuraamiseen. Päivittäisten palvelinvarmuuskopioiden avulla muutoksia voitiin tarvittaessa palauttaa jopa vuosien ajalta, mikä tarjosi varmuutta ja mahdollisti merkittävien muutosten tarkastelun. [6.]

Seuraava askel versiohallinnan kehittämisessä oli Microsoftin Team Foundation Serverin (TFS) käyttöönotto. TFS tarjosi joitakin edistyneempiä versiohallinnan ominaisuuksia, mutta yrityksessä ei silti ollut systemaattisia prosesseja, joilla olisi hallittu tehtäviä ja dokumentoitu muutoksia. Tehtävät ja työvaiheet olivat hajautettuina eri ympäristöihin, kuten Excel-tilukoihin ja OneNote-sivuille. Tämä hajanaisuus teki tehtävien ja muutosten seurannasta haastavaa, eikä muutosten dokumentaatiota ollut lainkaan. [5; 6.]

Ohjelmistovirheiden hallinta ja virheiden dokumentointi oli myös yksinkertaista ja epäjärjestelmällistä. Kastenilla käytettiin OneNotessa olevaa listaa, johon merkittiin havaittuja bugeja, ja ohjelmistosuunnittelijat yliviivasivat listalta korjatut virheet. Tämä prosessi ei tarjonnut kunnollista jäljitettävyyttä tai systemaattista tapaa arvioida bugien korjausten vaikutuksia tai varmistaa ohjelmiston laadun kehittymistä. Myöskään minkäänlaista systemaattista testausprosessia ei ollut käytössä, vaan bugien korjauksessa luotettiin siihen, että ohjelmoijat tekivät muutokset oikein ilman erillistä laadunvarmistusta. [6.]

Azure DevOpsin käyttöönotto on merkittävä askel Kastenin ohjelmistokehityksessä, sillä se tarjoaa keskitetyn alustan versiohallintaan, tehtävien hallintaan, dokumentointiin ja testaukseen. Tämän työkalun myötä yritys voi ensimmäistä kertaa hallita ohjelmistokehityksen prosesseja yhtenäisesti, seurata muutoksia systemaattisesti ja ottaa käyttöön jatkuvan integraation ja jatkuvan toimituksen kaltaisia DevOps-käytäntöjä, jotka tukevat laadunvarmistusta ja työn läpinäkyvyyttä.

Kirjoittajan rooli tässä työssä on ollut keskeinen Azure DevOps -ympäristön käyttöönotossa Kastenilla, ja siihen on sisältynyt dokumentointia, ympäristön konfigurointia ja eri ominaisuuksien tutkimista.

### 2.3 DevOps-perusteet ja Azure DevOpsin komponentit

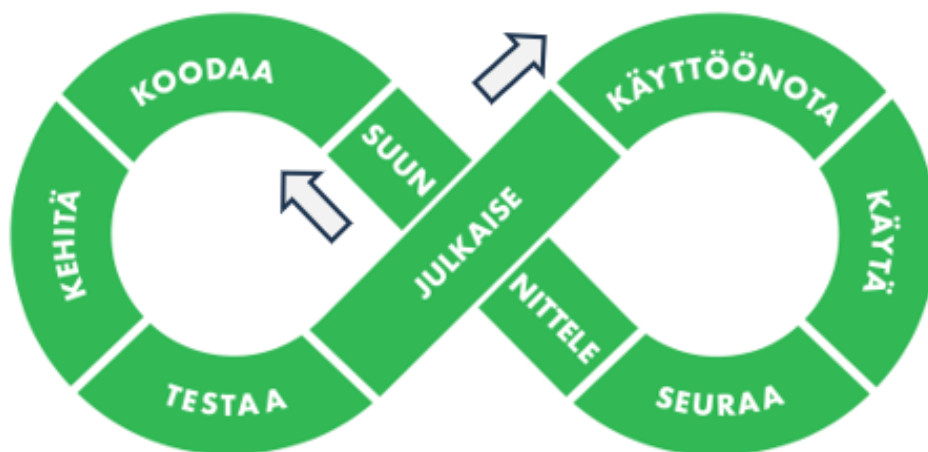
Azure DevOps on Microsoftin pilvipohjainen alusta, joka yhdistää projektinhallinnan, koodin hallinnan, testauksen ja CI/CD-putken (jatkuva integraatio ja jatkuva toimitus) hallinnan yhteen ympäristöön. Azure DevOps mahdollistaa kehitys- ja operatiivisten tiimien saumattoman yhteistyön ja tukee ohjelmistokehityksen tehokasta tiedonhallintaa. [7.]

DevOps-toimintamallin peruseriaatteiden ymmärtäminen ja Azure DevOpsin keskeisten komponenttien tunteminen auttavat hahmottamaan, miten nämä työkalut tukevat Kastenin tarpeita ohjelmistokehityksessä.

### 2.3.1 DevOpsin perusteet

DevOps (Development and Operations) on ohjelmistokehityksen toimintamalli, joka yhdistää ohjelmistokehityksen ja IT-operaatiot yhtenäiseksi prosessiksi. Sen tavoitteena on tehostaa ohjelmistotuotantoa ja lyhentää toimitusaikoja vähentämällä kehitys- ja käyttöönottoprosessin välistä kitkaa. DevOps-periaatteet pohjautuvat ketterän kehityksen käytäntöihin ja korostavat jatkuvaa integraatiota ja jatkuvaa toimitusta. Näiden avulla ohjelmistokehitys voi tapahtua iteratiivisesti ja nopeammin, mikä tukee nopeaa reagointia muutoksiin ja asiakkaiden tarpeisiin. [8.]

DevOpsia kuvataan usein päättymättömänä silmukkana, joka lähtee liikkeelle suunnittelusta ja etenee ohjelmointiin, koontiin, testaukseen, julkaisuun, käyttöönottoon, käyttöön, monitorointiin ja palautteen kautta jälleen suunnitteluun, josta silmukka lähtee uudelleen käyntiin kuvan 1 mukaisesti.



Kuva 1. DevOps kuvattuna päättymättömänä silmukkana [9].

DevOpsin keskeisimpiä tavoitteita ovat automaatio, jatkuva palaute ja tehokas tiedonhallinta. Automaatio mahdollistaa sen, että ohjelmiston testaus, integraatio ja käyttöönotto voidaan hoitaa automatisoidusti, mikä vähentää

manuaalisten virheiden määrää ja parantaa prosessien luotettavuutta. Jatkuva palaute kehitysprosessista auttaa tiimejä havaitsemaan ja ratkaisemaan ongelmia aikaisessa vaiheessa, mikä parantaa ohjelmiston laatua. Tehokas tiedonhallinta puolestaan mahdollistaa läpinäkyvän yhteistyön eri tiimien välillä, jolloin kehitys- ja operatiiviset tiimit voivat jakaa tietoa sujuvasti ja vähentää päällekkäistä työtä. [10.]

DevOps perustuu myös kulttuuriseen muutokseen organisaatioissa, sillä se vaatii uudenlaista ajattelutapaa ja tiimien välistä yhteistyötä. Perinteisesti kehitys- ja operatiiviset tiimit ovat toimineet erillään, mutta DevOpsin periaatteet yhdistävät nämä tiimit yhteisen tavoitteen – ohjelmiston jatkuvan parantamisen ja nopean julkaisun – ympärille. DevOps-kulttuuri korostaa avoimuutta, vastuullisuutta ja jatkuvaa parantamista, mikä mahdollistaa paremman sopeutumisen nopeisiin muutoksiin ja kasvaviin asiakasvaatimuksiin. [11.]

DevOps on kehittynyt merkittäväksi osaksi nykyaikaista ohjelmistokehitystä, erityisesti yrityksissä, jotka haluavat pysyä kilpailukykyisinä ja kehittää tuotteitaan nopeasti ja laadukkaasti. DevOpsin käytännöt ja työkalut, kuten Azure DevOps, tukevat organisaatioita näiden tavoitteiden saavuttamisessa, ja ne tarjoavat yrityksille mahdollisuuden automatisoida, seurata ja optimoida koko ohjelmistokehityksen elinkaarta. [1.]

### 2.3.2 Azure DevOps -komponentit

Azure DevOps tarjoaa seuraavat keskeiset komponentit, jotka tukevat DevOps-periaatteita:

- Azure Boards: Boards on tehtävien- ja projektinhallintatyökalu, joka mahdollistaa tehtävien luomisen, priorisoinnin ja seurannan. Tehtävät voidaan järjestää backlogeihin, sprintteihin ja epiceihin, mikä auttaa tiimejä suunnittelemaan ja seuraamaan työtään tehokkaasti. [12.]

- Azure Repos: Repos on Git-pohjainen versionhallintatyökalu, jonka avulla kehittäjät voivat tallentaa, jakaa ja hallita koodia keskitetysti. Repos tukee versiohallintaa, mikä parantaa tiimien yhteistyötä ja helpottaa koodin eri versioiden hallintaa. [13.]
- Azure Pipelines: Pipelines vastaa CI/CD-putkien hallinnasta. CI -vaihe automatisoi koodin integroinnin ja testauksen, kun taas CD -vaihe mahdollistaa jatkuvan julkaisun. Pipelines vähentää manuaalista työtä ja varmistaa, että tuotantoon päätyvä koodi on testattu ja julkaisukelpoinen. [14.]
- Azure Test Plans: Test Plans tarjoaa testauksen hallintatyökalun, joka mahdollistaa testitapausten luomisen ja automatisoinnin. Tämä työkalu tukee laadunvarmistusta osana CI/CD-prosessia, mikä auttaa varmistamaan, että jokainen ohjelmistopäivitys on testattu ennen tuotantoon siirtämistä. [15.]
- Azure Artifacts: Artifacts on riippuvuuksien hallintatyökalu, joka mahdollistaa pakettien hallinnan ja jakamisen tiimin kesken. Tämä vähentää päällekkäisyyksiä ja takaa, että kaikki riippuvuudet ovat saatavilla keskitetysti. [16.]

## 2.4 Käyttöönoton tavoitteet

Azure DevOpsin käyttöönotolla Kasten pyrkii ensisijaisesti keskittämään dokumentaation ja tehtävien hallinnan yhteen järjestelmään. Keskitetty ympäristö vähentää tarvetta käyttää useita eri työkaluja, mikä parantaa tiedon saatavuutta ja vähentää päällekkäistä työtä. Integraatiot muihin Microsoftin järjestelmiin, kuten Microsoft Teamsiin ja SharePointiin, tukevat tätä tavoitetta mahdollistamalla tiedon helpon jakamisen ja keskustelun suoraan työkalujen välillä. Esimerkiksi Teams-integraation kautta tiimit voivat kommunikoida ja seurata Azure DevOpsin tehtäviä reaaliajassa, ja SharePoint tukee

dokumentaation hallintaa, jolloin kaikki tärkeät tiedot ovat saatavilla yhdessä paikassa. [17.]

Toinen tärkeä tavoite on lisätä työn läpinäkyvyyttä. Azure Boardsin käyttö mahdollistaa tehtävien hallinnan ja priorisoinnin selkeällä tavalla, mikä parantaa työn näkyvyyttä ja helpottaa projektien seuranta. Tiimipalavereissa työtehtäviä ja projekteja voidaan käsitellä keskitetysti, jolloin keskusteluista syntyy digitaalinen muistijälki, joka tukee jatkuvaa kehitystä.

Tehokkuuden parantaminen automatisoinnilla on myös keskeinen tavoite. Azure Pipelinesin CI/CD-putket mahdollistavat koodin testauksen ja julkaisun automatisoinnin, mikä vähentää manuaalista työtä ja varmistaa laadukkaan koodin tuottamisen. Lisäksi Azure DevOpsin käyttöönotolla pyritään tehostamaan yhteistyötä ja varmistamaan hiljaisen tiedon tallentuminen. Kun dokumentaatio ja tehtävienhallinta keskitetään yhteen paikkaan, hiljaisen tiedon katoaminen vähenee, sillä tiedot ja keskustelut tallentuvat organisaation yhteiseen ympäristöön. Näin kaikki työntekijät pääsevät käsiksi samaan tietoon, mikä parantaa tiedonkulkua ja tukee eri tiimien välistä yhteistyötä.

Tavoitteiden saavuttaminen Azure DevOpsin avulla auttaa Kastenian yhtenäistämään prosessejaan, sujuvoittamaan yhteistyötä ja tehostamaan ohjelmistokehitysprosessiaan.

### **3 Tehtävien hallinta ja yhteistyön tehostaminen**

Tehtävien hallinta ja yhteistyön sujuvuus ovat keskeisiä tekijöitä tehokkaan ohjelmistokehityksen kannalta. Azure DevOps tarjoaa useita työkaluja, jotka tukevat tehtävien hallintaa ja tiimien välistä yhteistyötä. Kastenilla nämä työkalut auttavat parantamaan läpinäkyvyyttä ja työtehtävien priorisointia sekä mahdollistaneet hiljaisen tiedon näkyväksi tekemisen ja dokumentaation keskittämisen. Seuraavassa käsitellään keskeisiä toimintatapoja ja työkaluja, joilla tehtävien hallintaa ja yhteistyötä on tehostettu Kastenin organisaatiossa.

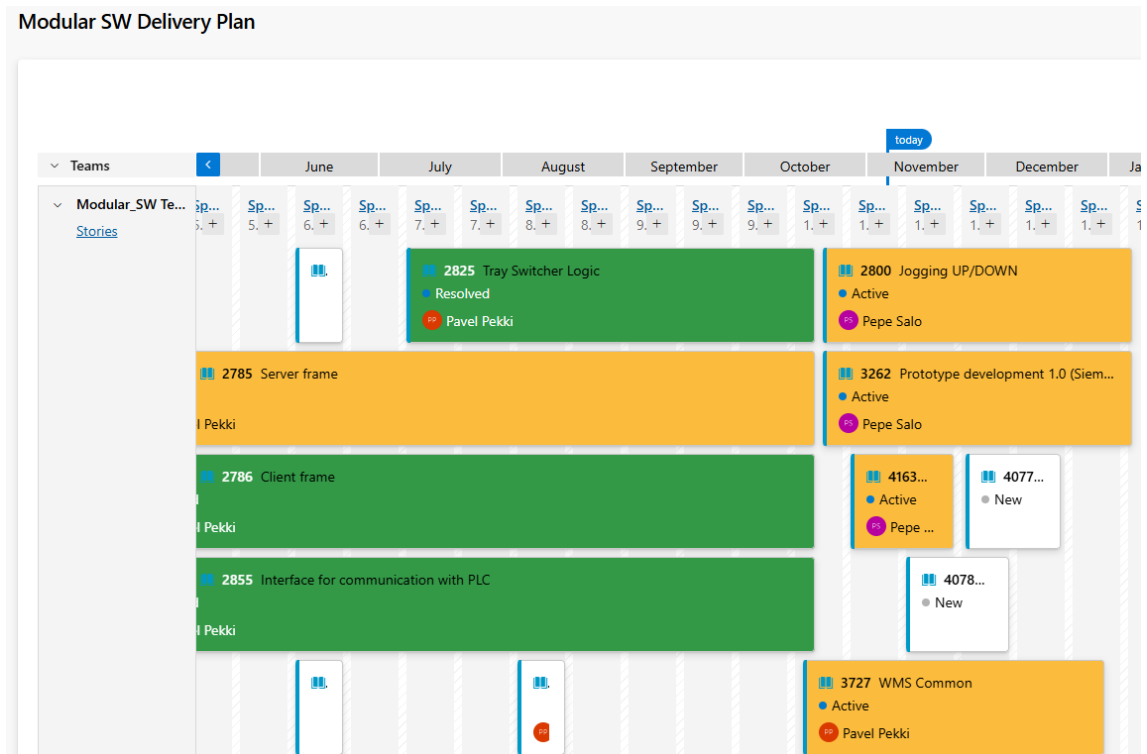
### 3.1 Azure Boards ja työnkulun seuranta

Kuten aiemmin kerrottiin, Azure Boards on projektihallintatyökalu, jonka avulla tiimit voivat luoda, seurata ja priorisoida tehtäviä selkeässä visuaalisessa ympäristössä. Kastenilla Azure Boards on keskeisessä roolissa, kun pyritään varmistamaan, että kaikki työtehtävät ovat näkyvillä ja helposti seurattavissa.

#### 3.1.1 Tehtävien hallinta ja priorisointi

Tehtävien hallinta ja priorisointi Azure Boardsissa mahdollistavat tehokkaan ja selkeän työnjaon. Kastenilla tehtävät jaotellaan eri tasoille, kuten backlogeihin ja sprintteihin, mikä auttaa tiimejä järjestämään työnsä tärkeysjärjestyksessä ja keskittymään kiireellisimpiin tehtäviin. Tehtävien jakaminen selkeisiin kokonaisuuksiin helpottaa myös resurssien kohdentamista ja mahdollistaa tiimien välisten vastuiden hallinnan. Tehtäville voidaan asettaa prioriteettitaso lisäksi vastuuhenkilöt ja aikataulut, mikä parantaa työn näkyvyyttä koko organisaatiossa. [18.]

Azure DevOpsin Delivery Plans -toiminto on ollut tärkeä työkalu Kastenin kuvassa 2 esitellyn uuden projektin aikataulutuksessa. Se tarjoaa selkeän aikajananäkymän, jonka avulla eri tiimit voivat nähdä omat tehtävänsä sprinttikohtaisesti ja seurata projektin etenemistä.



Kuva 2. Azure Boardsin Delivery Plan -toiminnon tuottama visualisointi projektin aikataulutuksesta.

Delivery Plans auttaa hallitsemaan riippuvuuksia eri työvaiheiden välillä, mikä mahdollistaa aikataulun tehokkaan optimoinnin ja vähentää viivästyksistä aiheutuvia pullonkauloja [19]. Lisäksi toiminnon avulla projektipäälliköt voivat reagoida nopeasti mahdollisiin muutoksiin ja varmistaa, että projekti etenee suunnitellusti, vaikka aikatauluun tulisi muutoksia.

### 3.1.2 Käyttäjätasojen hallinta

Azure DevOpsissa käyttäjätasojen hallinta on olennainen osa tehtävien ja tietojen hallinnan turvallisuutta ja tehokkuutta. Käyttäjät voidaan määrittää erilaisiin rooleihin, joilla on eri oikeudet, kuten lukuoikeudet, muokkausoikeudet tai järjestelmänvalvojan oikeudet [20]. Kastenilla tämä mahdollistaa sen, että työntekijät pääsevät käsiksi vain niihin tietoihin ja tehtäviin, jotka kuuluvat heidän vastualueelleen, mikä parantaa tietoturvaa ja vähentää virheiden riskiä.

Esimerkiksi tiimin jäsenet voivat muokata omia tehtäviään, mutta projektipäälliköt voivat hallita ja valvoa koko projektin tilannetta ja edistymistä.

Lisäksi Kasten hyödyntää palveluntarjoajia ohjelmointityössä, erityisesti PLC-ohjelmoinnin yhteydessä. Tällaisessa yhteistyössä käyttäjätasojen hallinta on erityisen tärkeää: palveluntarjoajille annetaan pääsy vain heidän tarvitsemiinsa tietoihin ja tehtäviin, mikä suojaa yrityksen sisäisiä tietoja ja tukee tehokasta yhteistyötä. Näin Kasten voi hallita palveluntarjoajien työskentelyä turvallisesti ja keskitetysti Azure DevOpsin kautta.

### 3.1.3 Tiimipalavereiden hyödyntäminen

Tiimipalavereissa Azure Boardsin käyttö on osoittautunut erittäin hyödylliseksi, sillä se mahdollistaa tehtävien tarkastelun ja päivittämisen reaaliaikaisesti. Palavereissa voidaan käydä läpi kunkin tehtävän edistyminen ja mahdolliset haasteet, mikä edistää avoimuutta ja tiimien välistä yhteistyötä. Tiimipalavereiden selkeät tehtävälisterit tukevat päätöksenteon nopeutta ja varmistavat tehtävien systemaattisen seurannan.

## 3.2 Hiljaisen tiedon näkyväksi tekeminen ja dokumentaation keskittäminen

Azure DevOpsin keskitetty alusta mahdollistaa hiljaisen tiedon, kuten kokemusten ja aiempien ratkaisujen, näkyväksi tekemisen ja tallentamisen koko organisaation hyödynnettäväksi. Ennen käyttöönottoa dokumentaatio oli hajautettu useisiin järjestelmiin, mikä teki tiedon hallinnasta haasteellista. Nyt dokumentaatio on kaikkien saatavilla yhdestä paikasta, ja se voidaan linkittää suoraan tehtäviin, projekteihin ja keskusteluihin [21]. Tämä vähentää tiedon katoamisen riskiä, erityisesti henkilöstömuutosten yhteydessä, ja tukee tehokasta tiedon jakamista.

### 3.3 Digitaalinen muistijälki

Azure DevOpsin käyttö on luonut Kastenille digitaalisen muistijäljen keskusteluista, tehtävistä ja dokumenteista. Tämä tarkoittaa, että kaikkiin projekteihin ja tehtäviin liittyvät päätökset ja keskustelut tallentuvat automaattisesti, jolloin niihin voidaan palata myöhemmin. Digitaalinen muistijälki vähentää tiedon katoamisen riskiä ja tekee aiempien päätösten ja prosessien tarkastelusta helppoa. Lisäksi digitaalinen muistijälki tukee jatkuvaa parantamista, kun tiimit voivat tarkastella aikaisempia projekteja ja ottaa opiksi aiemmista kokemuksista.

## 4 Prosessien optimointi ja laadunvarmistus

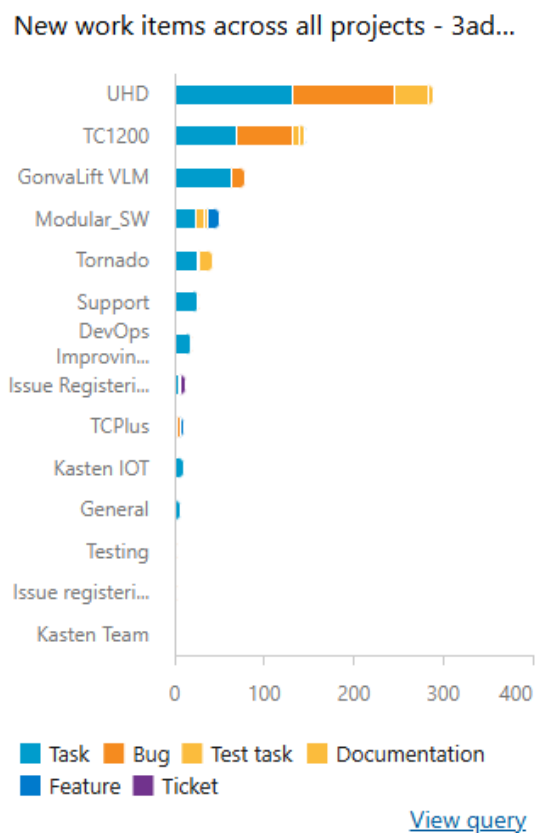
Prosessien optimointi ja laadunvarmistus ovat keskeisiä osa-alueita, joilla yritys voi parantaa kehitysprosessien tehokkuutta ja tuotteiden laatua. Tässä luvussa tarkastellaan, kuinka Kasten hyödyntää erilaisia analysointimenetelmiä ja automatisointityökaluja prosessien tehostamiseksi sekä laadunvarmistuksen tueksi. Luvussa käsitellään muun muassa tiedon visualisointia, ulkopuolisia työkaluja ja laadunvarmistuksen käytäntöjä, jotka kaikki yhdessä tukevat yrityksen kehitysprosessien parantamista ja tehokkuuden lisäämistä.

### 4.1 Queryt ja dashboardit

Queryjä eli hakukyselyitä käytetään tehtävien tilan, prioriteettien ja sprintin edistymisen tarkasteluun. Esimerkiksi avoimet ja valmiit tehtävät voidaan suodattaa prioriteetin mukaan, mikä auttaa tiimejä keskittymään kiireellisimpiin tehtäviin.

Dashboardit puolestaan tarjoavat visuaalisen yleiskuvan projektin tilasta reaaliaikaisesti. Ne yhdistävät queryistä saatavat tiedot ja esittävät reaaliajassa keskeiset mittarit, kuten tehtävien määrän, valmistumisasteen ja sprintin edistymisen. [22.] Dashboardit auttavat tiimejä tunnistamaan pullonkauloja ja varmistavat, että kaikki projektin jäsenet voivat helposti seurata työn

etenemistä. Kuvan 3 grafiikasta voi havaita, että yksi projekti työllistää reilusti muita enemmän myös Kastenilla.



Kuva 3. Kuva näyttää uusien työtehtävien määrän eri projekteissa jaottelemalla ne tehtävätyyppeihin, kuten tehtävät, bugit, testitehtävät, dokumentaatiot, ominaisuudet ja tiketit, mikä auttaa hahmottamaan projektien työtilannetta.

Näiden työkalujen avulla Kasten pystyy hallitsemaan tehtäviä ja resursseja tehokkaammin sekä reagoimaan nopeasti työn edistymisen muutoksiin. Queryt ja dashboardit tukevat Kastenin tiimejä varmistamaan projektien läpinäkyvyyden ja saavuttamaan tavoitteensa ajallaan.



Kuva 4. Kuvaaja esittää tehtävien kumulatiivista kasvua ja vähenemistä ajan myötä, mikä havainnollistaa projektin työkuormituksen kehitystä eri vaiheissa.

Azure DevOpsin query-työkalulla kerätyn datan voi avata Excelissä, missä tiedoista voi luoda graafisia esityksiä, kuten kuvassa 4 näkyvä tehtävien kumulatiivinen kasvu [23]. Graafinen esitys tehtävämäärän kehityksestä auttaa hahmottamaan työkuorman kasvua ja voi toimia perusteluna, jos projektin edetessä herää tarve lisäresurssien hankkimiselle.

## 4.2 Ulkopuoliset työkalut

Azure DevOpsin ulkopuoliset työkalut tarjoavat monia tapoja tehostaa prosesseja. Osa näistä työkaluista löytyy Azure DevOpsin omasta lisäosakaupasta, mutta jotkin toimivat täysin järjestelmän ulkopuolella ja vaativat käyttäjältä PAT -tunnistautumisen, jotta ne voivat tehdä muutoksia Azure DevOps -organisaatioon.

## 4.2.1 Power Automate

Power Automate on Microsoftin tarjoama työkalu, joka mahdollistaa työkulkujen automatisoinnin eri sovellusten ja palveluiden välillä ilman laajaa ohjelmointiosaamista [24]. Kastenilla Power Automate -työkalua on hyödynnetty luomaan automaattinen prosessi, joka käsittelee kehitysideoita ja palautteita suoraan sähköpostien perusteella. Tämä automatisointi parantaa huomattavasti palautteiden käsittelyn tehokkuutta, sillä saapuvat sähköpostiviestit muutetaan automaattisesti tehtäviksi Azure DevOps -ympäristössä, jolloin niitä voidaan hallita yhdessä paikassa muiden projektitehtävien kanssa.

Esimerkkikoodi 1 näyttää JSON-muotoisen Power Automate -skriptin, joka määrittää, miten Office 365 -sähköpostikysely suoritetaan. Skripti tarkkailee sähköpostilaatikkoo ja etsii uusia viestejä, jotka vastaavat määriteltäviä kriteerejä, kuten viestin lähettäjä ja liitteiden olemassaolo.

```
{
  "inputs": {
    "host": {
      "connectionName": "shared_office365",
      "operationId": "SharedMailboxOnNewEmailV2",
      "apiId": "/providers/Microsoft.PowerApps/apis/shared_of-
office365"
    },
    "parameters": {
      "mailboxAddress": "Feedback@kasten.fi;",
      "folderId": "Inbox",
      "from": "oskari.narhi@kasten.fi ",
      "importance": "Any",
      "hasAttachments": true,
      "includeAttachments": true
    },
    "authentication": "@parameters('$authentication')"
  },
  "recurrence": {
    "interval": 5,
    "frequency": "Minute"
  },
  "metadata": {
    "operationMetadataId": "9a514b0c-1053-40ce-892a-6ee5f753aa99"
  },
  "splitOn": "@triggerOutputs()?['body/value']"
}
```

Esimerkkikoodi 1. JSON-muotoinen Power Automate -skripti, joka määrittää, miten sähköpostikysely Office 365 -yhteydellä suoritetaan.

Kun sähköpostiviesti on tunnistettu ja valittu, Power Automate lähettää HTTP POST -pyynnön Azure DevOpsiin luodakseen uuden tehtävän viestin sisällön perusteella. Esimerkkikoodi 2 kuvaa JSON-pyyntöä, joka muuttaa sähköpostiviestin aihe tekstin, sisällön ja lähettäjän tiedot tehtävän kentiksi Azure DevOps -ympäristössä.

```
{
  "inputs": {
    "method": "POST",
    "uri": "*****",
    "headers": {
      "Authorization": "Basic *****",
      "Content-Type": "application/json-patch+json"
    },
    "body": [
      {
        "op": "add",
        "path": "/fields/System.Title",
        "from": null,
        "value": "@{triggerOutputs()?['body/subject']}"
      },
      {
        "op": "add",
        "path": "/fields/System.Description",
        "from": null,
        "value": "@{triggerOutputs()?['body/body']}"
      },
      {
        "op": "add",
        "path": "/fields/Custom.Stakeholder",
        "from": null,
        "value": "@{triggerOutputs()?['body/from']}"
      },
      {
        "op": "add",
        "path": "/fields/System.Tags",
        "from": null,
        "value": "Feedback"
      }
    ]
  },
  "metadata": {
    "operationMetadataId": "9b37ef8a-dea9-483f-b371-573372a6a825"
  }
}
```

Esimerkkikoodi 2. JSON-muotoinen skripti, joka kuvaa HTTP POST -pyyntöä, jolla taski luodaan sähköpostikyselyn perusteella.

## 4.2.2 PowerShell

Vaikka Azure DevOps tarjoaa monipuolisia toimintoja, kaikkea ei ole aina optimoitu käyttäjäystävällisyyden kannalta. Siksi Kastenilla käytetään myös PowerShell-skriptejä helpottamaan tiettyjä prosesseja. PowerShellin avulla voidaan automatisoida esimerkiksi sprinttien luominen Azure Boardsin Sprints-näkymälle, kuten esimerkkikoodissa 3 on esitelty.

```
$date: Get-Date -Year 2024 -Month 1 -Day 4
$sprintNumber = 1

while ($date.Year -ne "2025") {
    $Enddate: $date.AddDays(14)
    if ($EndDate.Year -eq "2025") {
        $Enddate: Get-Date -Year 2024 -Month 12 -Day 31
    }
    $week = Get-Date $date -UFormat "%V"

    $Sprint = @{
        Name = "Sprint $sprintNumber"
        ProjectName = 'UHD'
        Startdate: $date
        Finishdate: $EndDate
    }
    Add-UHD team @Sprint

    $date: $date.AddDays(14)
    $sprintNumber++
}
```

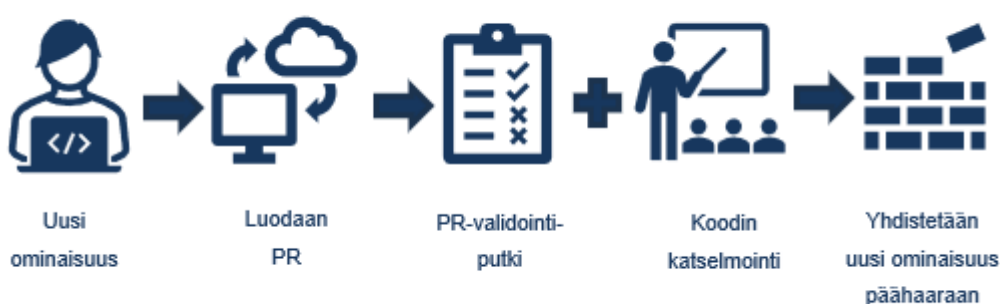
Esimerkkikoodi 3. Tämä PowerShell-koodi luo Kastenin UHD-projektiin kahden viikon (14 päivän) mittaisia sprinttejä, alkaen 4. tammikuuta 2024 ja jatkaen koko vuoden ajan.

Tulevaisuudessa PowerShell-skriptejä voidaan tulevaisuudessa hyödyntää entistä laajempaan ja moniulotteisempaan käyttötarkoitukseen, kun vireillä oleva CI/CD-putki saadaan luotua valmiiksi [25].

## 4.3 Laadunvarmistuksen nykyiset käytännöt

Kastenilla on käytössä useita laadunvarmistuskäytäntöjä, joiden avulla pyritään ehkäisemään virheitä ja parantamaan koodin laatua ennen sen päätymistä tuotantoon. Kuvassa 5 on esitetty laadunvarmistuksen vaiheet Azure

DevOpsissa. Keskeinen laadunvarmistuskäytäntö on koodikatselmointi, jossa jokainen koodimuutos tarkistetaan toisen tiimin jäsenen toimesta ennen sen hyväksymistä ja integrointia päähaaraan. Tämä katselmointi auttaa tunnistamaan mahdolliset virheet ja kehityskohteet varhaisessa vaiheessa. Lisäksi käytössä on Azure Reposin pull request (PR) -käytäntö, jossa kaikki muutokset dokumentoidaan ja hyväksytään ennen päähaaraan yhdistämistä, mikä tekee muutoksista jäljitettäviä ja mahdollistaa päähaaran palauttamisen aiempaan tilaan tarvittaessa.



Kuva 5. Uuden ominaisuuden tekeminen ohjelman päähaaraan Azure DevOpsin osalta.

Näiden vaiheiden lisäksi laadunvarmistusprosessiin kuuluu myös toinen ohjelmoijan suorittama muutosten testaus varastoautomaatilla. Tämä toisen henkilön suorittama testaus on tärkeä lisävarmistusvaihe, sillä se tarjoaa ulkopuolisen näkökulman, auttaa havaitsemaan mahdollisia ongelmia ja lisää turvallisuutta varastoautomaatin testausvaiheessa. Koneen moottoreiden suuri voima edellyttää erityistä varovaisuutta, mikä tekee manuaalisesta testauksesta tärkeän osan laadunvarmistusta.

Tämä yhdistelmä, jossa hyödynnetään koodikatselmointia, toisen ohjelmoijan suorittamaa testausvaihetta ja pull request -käytäntöä, parantaa koodin laatua, vähentää virheriskiä ja tukee turvallista ohjelmistokehitystä [26].

Säännöllisin väliajoin päähaarasta luodaan tuotantoon julkaistava versio, joka jaetaan mm. varastoautomaattien huoltoryhmille. Näin he voivat käyttää uusinta ohjelmistoversiota päivittäessään varastoautomaattien ohjelmistoja. Tämä käytäntö varmistaa, että huoltoryhmillä on aina käytettävissään ajantasainen ja laadunvarmistettu ohjelmistoversio, mikä helpottaa huoltotoimia ja tukee varastoautomaattien toimintavarmuutta.

#### 4.4 CI/CD-putkien suunnittelu ja automatisointi

CI/CD-putken päätavoitteena on mahdollistaa koodin nopea ja luotettava integrointi, testaus ja tuotantoon vienti automaattisesti [27]. CI/CD-putken käyttöönotto on keskeinen osa Kastenin DevOps-strategiaa, sillä se mahdollistaa nopean, laadukkaan ja automatisoidun ohjelmiston julkaisuprosessin. Kastenin CI/CD-putki on vielä kehitteillä, mutta sen suunnittelu ja toteutus kuuluvat yrityksen lähitulevaisuuden tavoitteisiin, ja ne tukevat ohjelmistokehityksen tehokkuutta ja laatua.

Kastenin suunnitelmana on rakentaa Pipeline-prosessi, joka yhdistää CI/CD-putken eri vaiheet. CI-vaiheessa koodi integroidaan useita kertoja päivässä yhteiseen repositorioon, mikä varmistaa sen ajantasaisuuden ja yhteensopivuuden muiden kehittäjien tekemien muutosten kanssa. Jokaisen koodimuutoksen yhteydessä käynnistyvät automatisoidut testit auttavat havaitsemaan ja korjaamaan virheet jo varhaisessa vaiheessa. [27.]

Kastenin tavoitteena CD-vaiheessa on automatisoida koodin siirtyminen testauksesta tuotantoon ilman manuaalisia väliportaita. Tämä vähentää tuotantoon päätyvien virheiden riskiä ja nopeuttaa ohjelmiston julkaisua. Automatisoitu julkaisu vähentää manuaalisen työn tarvetta ja vapauttaa kehittäjien resursseja muihin tehtäviin, mikä tukee Kastenin tehokkuustavoitteita ja varmistaa ohjelmiston korkeaa laatua. [27.] Kastenin suunnitelmissa on myös varmistaa CI/CD-putken skaalautuvuus siten, että järjestelmä tukee sekä pieniä päivityksiä että suurempia ohjelmistomuutoksia.

## 5 Lopputulokset, haasteet ja suositukset

Azure DevOpsin käyttöönotto Kastenilla on tuonut mukanaan merkittäviä parannuksia prosesseihin ja yhteistyöhön, mutta myös haasteita, jotka ovat vaatineet ratkaisuja ja sopeutumista. Tässä luvussa tarkastellaan näiden muutosten vaikutuksia Kastenin toimintaan sekä käydään läpi keskeiset opit, joita prosessista on saatu. Lisäksi esitetään suosituksia muille keskikokoisille yrityksille, jotka harkitsevat vastaavan järjestelmän käyttöönottoa.

### 5.1 Azure DevOpsin tuomat hyödyt Kastenille

Azure DevOpsin käyttöönoton on havaittu parantaneen tehtävien hallintaa ja työn seurannan läpinäkyvyyttä. Azure Boards -työkalun avulla tiimeille on luotu mahdollisuus jakaa ja priorisoida tehtäviä selkeästi, mikä on tehostanut työn organisointia ja vähentänyt päällekkäisyyksiä. Yhteiset tehtävälisterit ovat myös helpottaneet tiimipalaverien toteutusta ja parantaneet yhteistyötä, sillä kaikki tiimin jäsenet näkevät ajantasaisen tilanteen yhdessä paikassa.

Lisäksi dokumentaation keskittämisen yhteen järjestelmään on havaittu parantaneen tiedon saavutettavuutta ja vähentäneen tiedon katoamisen riskiä. Ennen Azure DevOpsin käyttöönottoa dokumentaatio oli hajautettuna eri järjestelmiin, mikä vaikeutti tiedon löytämistä ja jakamista. Keskitetyn dokumentaation on nähty tukevan hiljaisen tiedon tallentamista ja siirtämistä helposti saavutettavaan muotoon, mikä on osoittautunut erityisen arvokkaaksi organisaatiolle.

Laadunvarmistus on parantunut pull request -käytännön ja testausprosessien ansiosta, mikä on lisännyt koodin laadun varmistamisen tehokkuutta ja vähentänyt virheriskiä. Näin resurssit vapautuvat muihin kehitystehtäviin. Vaikka CI/CD-putki on vielä kehitysvaiheessa eikä sitä ole otettu laajasti käyttöön, sen suunnittelu etenee tukemaan automatisoitua testausta ja nopeampaa tuotantoon vientiä tulevaisuudessa.

Yhteistyö ja tiedon jakaminen eri tiimien välillä on myös havaittu tehostuneen. Azure DevOpsin luoman digitaalisen muistijäljen ansiosta yhteistyö erityisesti eri vastuualueilla työskentelevien henkilöiden välillä on helpottunut. Tämä digitaalinen muistijälki varmistaa, että kaikki projektiin liittyvä tieto on keskitetysti saavutettavissa, mikä on parantanut tiimien välistä kommunikaatiota ja tukenut tavoitteiden saavuttamista.

## 5.2 Käyttöönottoprosessin haasteet ja ratkaisut

Azure DevOpsin käyttöönotossa on havaittu myös haasteita. Ensimmäinen haaste on liittynyt oppimiskäyrään ja muutosvastarintaan. Koska Azure DevOps on monipuolinen työkalu, sen laajat ominaisuudet ovat aiheuttaneet osalle käyttäjistä haasteita, erityisesti niille, joilla ei ole ollut aikaisempaa kokemusta projektinhallinnan tai koodinhallinnan työkaluista. Nämä ovat tyypillisiä haasteita monille organisaatioille, jotka pyrkivät ottamaan käyttöön vastaavanlaisia DevOps-työkaluja [28]. Ratkaisuna tähän on järjestetty koulutuksia ja tarjottu tukimateriaaleja, jotka ovat helpottaneet uuden järjestelmän omaksumista.

Toinen merkittävä haaste on ollut resurssien ja ajan rajallisuus. DevOps-järjestelmän täysimittainen käyttöönotto vaatii merkittävästi resursseja ja ajankäyttöä, erityisesti alkuvaiheessa, kun järjestelmää konfiguroidaan ja prosesseja muokataan vastaamaan organisaation tarpeita. Tämä on pyritty ratkaisemaan toteuttamalla käyttöönotto vaiheittain, jolloin tiimit voivat sopeutua työkaluihin asteittain ja työkuormaa saadaan tasattua.

Käyttöönottoprosessissa on kohdattu myös haasteita prosessien mukauttamisessa Azure DevOpsin tarjoamiin työkaluihin. Esimerkiksi tehtävien hallintatavat ja projektien raportointikäytännöt ovat vaatineet hienosäätöä, jotta ne vastaavat organisaation erityistarpeita. Tämä jatkuva mukauttaminen ja prosessien hienosäätö on nähty välttämättömänä, jotta Azure DevOps palvelee organisaatiota parhaalla mahdollisella tavalla.

Käyttöönoton yhteydessä on myös havaittu jatkuvan kehittämisen ja palautteen hyödyntämisen merkitys. Palautetta on kerätty käyttäjiltä, jotta DevOps-ympäristöä voidaan mukauttaa ja kehittää edelleen vastaamaan paremmin tiimien tarpeita ja tavoitteita. Palautteen hyödyntämisen on todettu olevan keskeinen tekijä käyttöönoton onnistumiselle, ja tätä käytäntöä on tarkoitus jatkaa myös tulevaisuudessa.

### 5.3 Käyttäjäpalaute ohjelmistokehityksen ulkopuolella

Azure DevOpsin käyttöönotto Kastenilla on saanut vaihtelevaa palautetta, erityisesti ohjelmistokehityksen ulkopuolisilta käyttäjiltä. Käyttäjäpalaute on ollut vähintäänkin ristiriitaista, ja yleisellä tasolla hieman negatiivista. Moni käyttäjä kokee, että uusi järjestelmä lisää heidän työtaakkaansa, sillä heillä on jo ennestään käytössään muita alustoja ja työkaluja eri työvaiheiden ja projektien hallintaan. Tämä on johtanut siihen, että osa käyttäjistä näkee Azure DevOpsin päällekkäisenä järjestelmänä eikä täysin välttämättömänä heidän tehtäviensä kannalta.

Lisäksi käyttäjät kokevat, että useiden alustojen käyttö hankaloittaa tiedonhallintaa ja vaatii lisää aikaa eri järjestelmiin kirjautumiseen ja tietojen etsimiseen. Tämä saattaa vaikuttaa negatiivisesti heidän kokemukseensa järjestelmän hyödyllisyydestä ja tarkoituksenmukaisuudesta. Palautteen perusteella onkin tärkeää arvioida, voitaisiinko Azure DevOpsia mukauttaa siten, että se palvelisi paremmin myös ohjelmistokehityksen ulkopuolisten käyttäjien tarpeita ja vähentäisi päällekkäisyyksiä muiden järjestelmien kanssa.

### 5.4 Lopputulokset ja opit

Kastenin kokemus Azure DevOpsin käyttöönotosta tarjoaa useita arvokkaita oppeja. Yksi keskeisistä lopputuloksista on ollut tehtävien hallinnan ja yhteistyön tehostuminen. Azure Boardsin ansiosta kaikki työtehtävät ja projektit ovat näkyvissä yhdessä ympäristössä, mikä on helpottanut tehtävien priorisointia ja tukenut tiimien välistä yhteistyötä. Tämä kokemus osoittaa, että

keskitetty tehtävienhallinta voi merkittävästi parantaa tiimien tehokkuutta ja vähentää kommunikaatioon liittyviä ongelmia.

Toinen merkittävä oppi liittyy dokumentaation keskittämiseen ja tiedon saavutettavuuden parantamiseen. Azure DevOpsin avulla Kasten on onnistunut siirtämään kaiken dokumentaation yhteen paikkaan, mikä on lisännyt tiedon löydettävyyttä ja vähentänyt tietojen päällekkäisyyttä. Keskitetty dokumentaatio on osoittautunut erityisen tärkeäksi organisaatiossa, jossa tiedot olivat aiemmin hajautettuina useisiin eri työkaluihin.

Kolmas tärkeä oppi on versionhallinnan parantuminen Azure Reposin avulla. Azure Repos on tuonut koodiversioiden hallintaan keskitetyn ratkaisun, joka helpottaa muutosten seuranta ja takaa, että tiimit työskentelevät ajantasaisen koodin parissa.

Vaikka CI/CD-putki ei ole vielä täysin käytössä, sen suunnittelu on tarjonnut arvokkaita näkemyksiä automaation hyödyistä laadunvarmistuksessa. Automatisoinnin mahdollisuudet ovat selvästi parantaneet ohjelmistokehityksen laatua ja tuoneet prosessiin tehokkuutta. Lopuksi, käyttöönoton haasteet, kuten oppimiskäyrä ja muutosvastarinta, ovat osoittaneet, että uuden työkalun menestyksellä käyttöönotto vaatii riittävää koulutusta, vaiheittain etenevää siirtymää sekä palautteen jatkuvaa hyödyntämistä.

## 5.5 Suositukset keskikokoisille yrityksille

Kastenin kokemukset Azure DevOpsin käyttöönotosta antavat hyödyllisiä suosituksia muille keskikokoisille yrityksille, jotka harkitsevat vastaavanlaisen DevOps-ratkaisun käyttöönottoa. Ensimmäiseksi on suositeltavaa aloittaa käyttöönotto vaiheittain ja keskittyä aluksi niihin ominaisuuksiin, jotka tuovat nopeimmat ja konkreettisimmat hyödyt, kuten tehtävienhallinnan ja dokumentaation keskittämisen.

Toiseksi, riittävän koulutuksen ja käyttäjätuen merkitys on keskeinen erityisesti niille käyttäjille, jotka eivät ole ohjelmistokehityksen ammattilaisia. Heille voi olla tarpeen tarjota yksinkertaistettuja ohjeita ja käytännön esimerkkejä työkalun hyödyistä, mikä helpottaa järjestelmän omaksumista.

Kolmantena suosituksena on mukauttaa DevOps-ympäristö yrityksen tarpeisiin. Azure DevOps tarjoaa monipuolisia toimintoja, mutta kaikkia ominaisuuksia ei välttämättä tarvita. On tärkeää valita vain ne työkalut ja prosessit, jotka tukevat yrityksen tavoitteita, ja räätälöidä ympäristö organisaation erityistarpeiden mukaan.

Lopuksi, jatkuva palautteenkeruu ja järjestelmän kehittäminen käyttäjien tarpeiden mukaan ovat keskeisiä DevOpsin menestykselliselle käytölle. DevOps on dynaaminen prosessi, joka vaatii jatkuvaa arviointia ja mukauttamista. Käyttöönoton jälkeen on tärkeää kerätä säännöllisesti palautetta ja mukauttaa järjestelmää sen pohjalta.

## **6 Yhteenveto**

Tämän insinööriyön tavoitteena oli tutkia Azure DevOps -ympäristön käyttöönottoa keskikokoisessa yrityksessä ja arvioida, kuinka hyvin se vastaa Kastenin kaltaisen organisaation tarpeisiin. Työssä analysoitiin käyttöönoton hyötyjä, haasteita ja mahdollisuuksia sekä esiteltiin DevOpsin keskeisiä käsitteitä ja työkaluja. Lisäksi esiteltiin Azure DevOpsin eri osa-alueiden, kuten tehtävienhallinnan, dokumentaation keskittämisen ja CI/CD-putken vaikutuksia yrityksen ohjelmistokehityksen työnkulkuun ja yhteistyöhön.

Kohdeyrityksen tarpeet huomioiden työssä keskityttiin erityisesti siihen, miten Azure DevOps voi parantaa tiedonhallintaa, yhteistyötä ja työn seurattavuutta. Käyttöönotto Kastenilla aloitettiin tarpeesta keskittää aiemmin hajautettu dokumentaatio ja tehtävienhallinta yhteen paikkaan, ja työssä kuvattiin, miten Azure DevOpsin työkalut auttavat saavuttamaan tämän tavoitteen.

Johtopäätöksenä voidaan todeta, että Azure DevOps tarjoaa keskikokoisille yrityksille, kuten Kastenille, tehokkaita työkaluja ja prosesseja ohjelmistokehityksen hallintaan. Työssä havaittiin, että Azure DevOpsin käyttö tehosti tehtävien hallintaa, paransi yhteistyön näkyvyyttä ja lisäsi tiedon saavutettavuutta koko organisaatiossa. Toisaalta käyttöönotto toi mukanaan myös haasteita, kuten oppimiskäyrän ja muutosvastarinnan, jotka kuitenkin pystyttiin osittain ratkaisemaan koulutuksen ja vaiheittaisen käyttöönoton avulla.

Tämä työ tarjoaa Kastenin kokemusten pohjalta suosituksia muille keskikokoisille yrityksille, jotka harkitsevat DevOpsin käyttöönottoa. Näihin suosituksiin kuuluu muun muassa vaiheittainen käyttöönotto, koulutuksen merkitys ja palautteen hyödyntäminen järjestelmän kehittämisessä. Työn perusteella voidaan myös todeta, että Azure DevOps soveltuu hyvin keskikokoisten yritysten tarpeisiin, ja se tarjoaa joustavan ja keskitetyn ratkaisun, joka tukee organisaation tavoitteita ja tehostaa ohjelmistokehityksen prosesseja.

## Lähteet

- 1 Slingerland, Cody. 55 Fascinating DevOps Statistics You NEED To Know In 2024. Verkkoaineisto. <https://www.cloudzero.com/blog/devops-statistics/>. Luettu 8.11.2024.
- 2 GitLab. Verkkoaineisto. <https://about.gitlab.com/topics/devops/> Luettu 7.11.2024.
- 3 Kasten. Verkkoaineisto. <https://www.kasten.fi/>. Luettu 5.10.2024.
- 4 Kasten. Verkkoaineisto. <https://www.kasten.fi/Yritys/Meista/Historia/>. Luettu 8.11.2024.
- 5 Salo, Pepe 2024. Constructor Finland Oy. Tekninen johtaja. Suullinen haastattelu 8.11.2024. Muistiinpanot kirjoittajan hallussa.
- 6 Pekki, Pavel 2024. Constructor Finland Oy. Ohjelmistokehityksen johtaja. Suullinen haastattelu 8.11.2024. Muistiinpanot kirjoittajan hallussa.
- 7 Microsoft. Azure DevOps dokumentaatio. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops&toc=%2Fazure%2Fdevops%2Fget-started%2Ftoc.json>. Luettu 28.9.2024.
- 8 5 Key DevOps principles. Verkkoaineisto. <https://www.atlassian.com/devops/what-is-devops>. Luettu 7.11.2024.
- 9 Suunnittelu, kehittäminen ja validointi. Verkkoaineisto. <https://btmalli.fi/book/development/design-development-and-validation/> Luettu 7.11.2024.
- 10 Hall, Tom. DevOps Best Practices. Verkkoaineisto. <https://www.atlassian.com/devops/what-is-devops/devops-best-practices>. Luettu 9.11.2024.
- 11 Hall, Tom. What is DevOps Culture? Verkkoaineisto. <https://www.atlassian.com/devops/what-is-devops/devops-culture>. Luettu 9.11.2024.
- 12 What is Azure Boards? Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops>. Luettu 6.11.2024.

- 13 What is Azure Repos? Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos?view=azure-devops>. Luettu 6.11.2024.
- 14 What is Azure Pipelines? Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>. Luettu 6.11.2024.
- 15 What is Azure Test Plans? Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/test/overview?view=azure-devops>. Luettu 6.11.2024.
- 16 Start using Azure Artifacts. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/artifacts/start-using-azure-artifacts?view=azure-devops&tabs=nuget%2Cnugetserver%2Cnugetserver19>. Luettu 6.11.2024.
- 17 Use Azure Boards in Microsoft Teams. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/boards/integrations/boards-teams?view=azure-devops>. Luettu 7.11.2024.
- 18 Plan and track work in Azure Boards. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/boards/get-started/plan-track-work?view=azure-devops&tabs=agile-process>. Luettu 7.11.2024.
- 19 Review team delivery plans in Azure Boards. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/boards/plans/review-team-plans?view=azure-devops>. Luettu 7.11.2024.
- 20 Default permissions and access levels for Azure Boards. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/boards/get-started/permissions-access-boards?view=azure-devops>. Luettu 7.11.2024.
- 21 Create a wiki for your project. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/project/wiki/wiki-create-repo?view=azure-devops&tabs=browser>. Luettu 7.11.2024.
- 22 Add, rename, and delete dashboards in Azure DevOps. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/devops/report/dashboards/dashboards?view=azure-devops>. Luettu 8.11.2024.
- 23 Use Excel to visualize data from the web UI. Verkkoaineisto. <https://learn.microsoft.com/en-us/azure/data-explorer/excel?tabs=azure-data-explorer>. Luettu 9.11.2024.

- 24 Mikä ihmeen Power Platform? Verkkoaineisto.  
<https://sulava.com/liiketoiminnan-digitalisointi-tiedolla-johtaminen/mika-ihmeen-power-platform/> Luettu 10.11.2024.
- 25 PowerShell scripts to customize pipelines. Verkkoaineisto.  
<https://learn.microsoft.com/en-us/azure/devops/pipelines/scripts/powershell?view=azure-devops&tabs=yaml> Luettu 9.11.2024.
- 26 SmartBear. Best Practices for Code Review.  
<https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>. Luettu 9.11.2024.
- 27 Bigelow, Stephen J. CI/CD pipelines explained: Everything you need to know <https://www.techtarget.com/searchsoftwarequality/CI-CD-pipelines-explained-Everything-you-need-to-know>. Luettu 9.11.2024.
- 28 Lipnitski, Andrei. How to Overcome Resistance to DevOps Implementation. Verkkoaineisto. <https://devops.com/how-to-overcome-resistance-to-devops-implementation/>. Luettu 9.11.2024.