

Henrik Korpi

Coursera sertifikaatti Tietojenkäsittelyn tutkinto-ohjelman osana

Päiväkirjamuotoinen opinnäytetyö Coursera sertifikaatin suorittamisesta

Coursea sertifikaatti Tietojenkäsittelyn tutkinto-ohjelman osana

Päiväkirjamuotoinen opinnäytetyö Coursea sertifikaatin suorittamisesta

Henrik Korpi
Opinnäytetyö
Syksy 2023
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma

Tekijä: Henrik Korpi

Opinnäytetyön nimi: Coursera sertifikaatti Tietojenkäsittelyn tutkinto-ohjelman osana

Työn ohjaaja: Teppo Räisänen

Työn valmistumislukukausi ja -vuosi: Syksy 2023

Sivumäärä: 44

Tämä opinnäytetyö on päiväkirjamuotoinen raportti ja se kertoo IBM DevOps and Software Engineering Specialization -sertifikaattien suorittamisen eri vaiheet. Päiväkirjaosio on jaettu 14 osaan, sertifikaatin sisältämien kurssien mukaan. Työtä esitellään kurssikohtaisesti keskittyen kurssin tehtäviin, haasteisiin, teemaan ja yhteenvetoon. Päiväkirjan lisäksi opinnäytetyö sisältää kirjoittajan nykytilanteesta kertovan osion, tietoperustan temalle ja pohdinnan.

Opinnäytetyön tavoitteena on pohtia Coursera sertifikaatin suorittamista Tietojenkäsittelyn tutkinto-ohjelman osana. Tämän lisäksi opinnäytetyön tavoite on tukea kirjoittajan kehittymistä sovelluskehittäjänä ja syventää ymmärrystä DevOps kehitysmallin mukaisesta työskentelystä. Päiväkirjamuotoinen opinnäytetyö antaa kuvan siitä, miten Coursera sertifikaatit toimivat, mitä niiden tekeminen vaatii ja miten niiden suorittaminen on toteutettu.

Valittu teema liittyy sertifikaatin kursseihin ja sen tietoperustana käytetään muun muassa tunnettujen DevOps kehittäjien kirjoituksia, artikkeleita ja teknistä dokumentaatiota.

Opinnäytetyön lopussa on pohdinta, jonka ensimmäisenä näkökulmana käydään läpi sertifikaatin suorittaminen osana opintoja ja mitä lisäarvoa se antaa Tietojenkäsittelyn tutkinto-ohjelmalle. Toisena näkökulmana kirjoittaja pohtii mitä hyötyä sertifikaatin suorittamisesta oli hänelle itselleen.

Opinnäytetyön tulos tukee vahvasti Coursera sertifikaatin suorittamista osana opintoja. Sertifikaatin työelämälähtöisyys ja aihe antavat lisäarvoa Tietojenkäsittelyn tutkinto-ohjelmalle ja mahdollistaa erikoistumisen aiheeseen opiskelijalle.

Asiasanat: DevOps, sovelluskehitys, sertifikaatti, ohjelmiston elinkaari, päiväkirja

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author: Henrik Korpi

Title of thesis: Coursera certificate as a part of Business Information Systems studies

Supervisor: Teppo Räisänen

Term and year when the thesis was submitted: Fall 2023

Number of pages: 44

This thesis is a diary style report of completing IBM DevOps and Software Engineering Specialization -certification through Coursera platform and contemplating the benefits and value of Coursera certificates in Business Information Systems studies. The diary section is divided into 14 parts, each covering a different course required for the completion of the certification. The thesis also includes a section about the writer's current situation, theory section for selected theme and a section for summary and reflection.

The main point of the thesis is to support the writer's professional development in software development and DevOps practices and to present a completion of a Coursera certificate. In addition, the benefits and value of completing a certificate during studies is evaluated.

The theory section for themes is sourced from known DevOps practitioners' writings, articles and technical documentation. The themes were chosen to cover the most important parts of DevOps practices.

Last section of the thesis summarizes the diary and contemplates on the benefits of completing the certificate for your studies and what benefits and value it provides to students and school alike.

Results support the use of Coursera certificate as a part of Business Information Systems studies for its added value to the degree and offering specialization in studies for students.

Keywords: DevOps, software development, certificate, software development life cycle, diary

SISÄLLYS

1	JOHDANTO	6
2	KÄSITTEET	7
3	NYKYTILANTEEN KUVAUS	9
4	DEVOPS.....	11
4.1	Mitä DevOps on?.....	11
4.2	DevOpsin tausta.....	12
4.3	DevOpsin käyttö.....	13
4.4	DevOps kehitysmalli.....	14
4.5	DevOps työkalut.....	16
5	PÄIVÄKIRJA.....	18
5.1	Kurssi 1: Introduction to DevOps.....	18
5.2	Kurssi 2: Introduction to Cloud Computing	20
5.3	Kurssi 3: Introduction to Agile Development and Scrum.....	22
5.4	Kurssi 4: Introduction to Software Engineering.....	23
5.5	Kurssi 5: Hands-on Introduction to Linux Commands and Shell Scripting.....	24
5.6	Kurssi 6: Getting Started with Git and GitHub	26
5.7	Kurssi 7: Python for Data Science, AI & Development	27
5.8	Kurssi 8: Python Project for AI & Application Development.....	29
5.9	Kurssi 9: Introduction to Containers w/ Docker, Kubernetes & Openshift.....	30
5.10	Kurssi 10: Application Development using Microservices and Serverless.....	32
5.11	Kurssi 11: Introduction to Test Driven Development.....	33
5.12	Kurssi 12: Continuous Integration and Continuous Delivery.....	35
5.13	Kurssi 13: Application Security and Monitoring	36
5.14	Kurssi 14: DevOps Capstone Project	38
6	POHDINTA.....	41
	LÄHTEET.....	43

1 JOHDANTO

Tämän päiväkirjamuotoisen opinnäytetyön tarkoitus on pohtia Coursera sertifikaatin suorittamista Tietojenkäsittelyn tutkinto-ohjelman osana ja esittää IBM DevOps and Software Engineering Specialization sertifikaatin suorittaminen ja arvioida sen hyötyä kirjoittajalle. Sertifikaatti rakentuu 14 kurssin ympärille, jotka kokonaisuutena antavat perustason ymmärryksen sovelluskehityksestä DevOps kehitysmallin mukaisesti. Päiväkirja on jaettu 14 eri osaan kurssien mukaan ja osissa kuvataan kurssien suorittamista, tehtäviä ja haasteita, sekä yhteenvedossa peilaan kurssin sisältöä teeman pohjalta.

Valitsin tämän aiheen suorittaessani puolen vuoden harjoittelujaksoa ison finanssi- ja vakuutusalan yrityksen sovelluskehitysosastolla. Minulla ei ollut tätä ennen kokemusta työskentelystä sovelluskehityksen parissa, ja ero opiskelun ja työelämän välillä oli suuri. Yrityksen sovelluskehityksyksikössä tehdään töitä DevOps kehitysmallin mukaisesti ja pidin paljon laaja-alaisesta työnkuvasta ja tavasta työskennellä. Teemoja on vain yksi, koska DevOps on erittäin laaja käsite. DevOps on ketterään kehitykseen, jatkuvaan integraatioon, jatkuvaan julkaisuun ja tuotantolinjojen automatisaatioon pyrkivä malli. Se on kokoelma erilaisia käytäntöjä, työkaluja ja kulttuurillista filosofiaa, joka yhdistää kehityksen ja tuotannon prosesseja automaation avulla. Teoriaosuudessa käytän tietoperustana teknistä dokumentaatiota, DevOps kehittäjien kirjoituksia ja artikkeleita.

Kouluni on ottanut Courseran käyttöön vuoden 2023 alusta. Koska Coursera on uusi lisäys koulutustarjontaan, sen tarjoamasta hyödyistä ei ole paljoa kokemusta tai tietoa. Opiskelijat voivat maksutta suorittaa Tietojenkäsittelyn tutkinto-ohjelmaan liittyviä sertifikaatteja ja hyväksilukea niitä osaksi opintoja. Sertifikaatit ovat työelämälähtöisiä ja niitä tarjoavat isot yritykset mm. IBM, Google ja Meta. IBM DevOps and Software Engineering Specialization-sertifikaatin kurssit sisältävät oppimismateriaaleina videotallenteita, tekstiä ja labroja, jotka suoritetaan virtuaalioasemalla. Kurssit arvostellaan monivalintatestein ja palautettavien töiden pohjalta. Kurssien suorittamiseen tarvitsee vain tietokoneen Internet-yhteydellä, erillisiä ohjelmia ei koneeseen tarvitse asentaa. Sertifikaatit ovat tarjolla ainoastaan englannin kielellä.

2 KÄSITTEET

Back-end – Käyttäjälle näkymätön osa toiminnallisuudesta sovelluskehityksessä, kuten tietokantayhteydet ja tiedon käsittely.

CI/CD-putki (Continuous Integration/Continuous Delivery) – Jatkuva integraatio ja julkaisu. Automatisoitu vaihe, jonka läpi koodi viedään versionhallintaan.

DevOps – Kehitysmalli, jonka periaatteita ovat ketterä kehitys, CI/CD ja automatisaatio, sekä yhteistyö kehityksen ja tuotannon kanssa.

Elinkaari – Prosessi, joka kattaa sovelluksen tekemisen eri vaiheet kokonaisvaltaisesti alusta loppuun.

Flask – Pythonilla luotu kevyt viitekehys verkkosovellusten tekoon.

Front-end – Käyttäjälle näkyvä osa toiminnallisuudesta sovelluskehityksessä, kuten käyttöliittymä.

Full-stack – Kattaa niin näkyvän kuin näkymättömän osan toiminnallisuudesta sovelluskehityksessä.

Git – Hajautettu versionhallintajärjestelmä sovellusten lähdekoodin hallintaan.

IaC (Infrastructure as Code) – Infrastruktuurin hallinnointi koodin avulla.

Ide (Integrated development environment) – Ohjelmointityökalu, joka tarjoaa käyttöliittymän koodaukseen.

Iteraatio – Kehitysmallin läpikäymiseen varattu aika.

Skripti – Automatisoitu setti tehtäviä, jonka voi suorittaa.

Repository – Lähdekoodin säilytyspaikka versionhallinnassa.

TDD (Test Driven Development) – Kehitysmenetelmä, jossa ensiksi kirjoitetaan testit ja testien pohjalta koodi.

Työnkulku (Workflow) – Yhdistää sovelluskehityksen eri vaiheet toisiinsa.

3 NYKYTILANTEEN KUVAUS

Olen kolmannen vuoden opiskelija Oulun Ammattikorkeakoulussa, Tietojenkäsittelyn tutkinto-ohjelmassa ja opintoni ovat loppusuoralla. Aloitin kuuden kuukauden harjoittelun ison finanssi- ja vakuutusalan yrityksen sovelluskehitysyksikössä tammikuusta 2023. Harjoitteluni pitää sisällään yleis- ja ammattiharjoittelun, sekä toimeksiannon suorittamisen harjoitteluna. Suunnittelin opintoni mahdollistamaan pitkän yhtäjaksoisen harjoittelun, koska koen sen antavan enemmän arvoa niin minulle opiskelijana, kuin yritykselle, joka käyttää resursseja minun kouluttamiseeni. Harjoittelussani työskentelen back-end puolella käyttäen Javaa. En voi avata harjoitteluani tämän enempää salassapitosopimuksen takia.

Vuosikurssillani ei ole erillistä suuntautumisvaihtoehtoa, vaan suuntautuminen on muodostunut henkilökohtaisen opintosuunnitelman suunnittelusta ja hyväksyttämisestä koulun kanssa. Vapaus opintojen suunnittelussa mahdollisti sovelluskehitykseen suuntautumisen. Minua on aina kiinnostanut kokonaisvaltainen sovelluskehitys ja sen kanssa käytetyt työmallit. Olen työskennellyt 15 vuotta rakennustekniikan parissa ja lineaarisesti kulkevat kankeat projektit ovat tulleet liian tutuiksi. Opintojen ja harjoittelun aikana olen huomannut nauttivani ketterän kehitysmallin ja DevOpsin mukaisesta joustavasta tiimityöskentelystä ja avoimuudesta työnteossa. Full-stack sovelluskehitys on mielenkiintoista laaja-alaisuutensa vuoksi ja ongelmienratkonta back-end puolella tuntuu erittäin mielekkäältä.

Kouluni otti vuoden 2023 alusta käyttöön Courseran career-academy palvelun. Coursera on maailmanlaajuinen verkko-opiskelualusta, joka tarjoaa englanninkielisiä kursseja suurten yliopistojen ja yritysten tarjoamana. Kursseja päivitetään jatkuvasti vastaamaan kehityksen ja työelämän vaatimuksia. Career-academy palvelu tarjoaa erillisistä kursseista koottuja kokonaisuuksia, joiden suorittamisesta myönnetään sertifikaatti. Kouluni pystyy career-academyn kautta tarjoamaan opiskelijalle ajantasaisen ja laadukkaan lisän Tietojenkäsittelyn tutkinto-ohjelmaan. Career-academyn kokonaisuudet tarjoavat mahdollisuuden erikoistua valittuun aiheeseen.

Olin erittäin innoissani kuultuani, että kouluni otti Coursera-sertifikaatit valikoimiinsa. Olen yrittänyt pitää tiukkaa tahtia opintojen edistymisen suhteen ja vapaasti Internetissä suoritettavat työelämälähtöiset sertifikaatit kiinnittivät heti huomioni. Koen saavani kaksinkertaisen hyödyn –

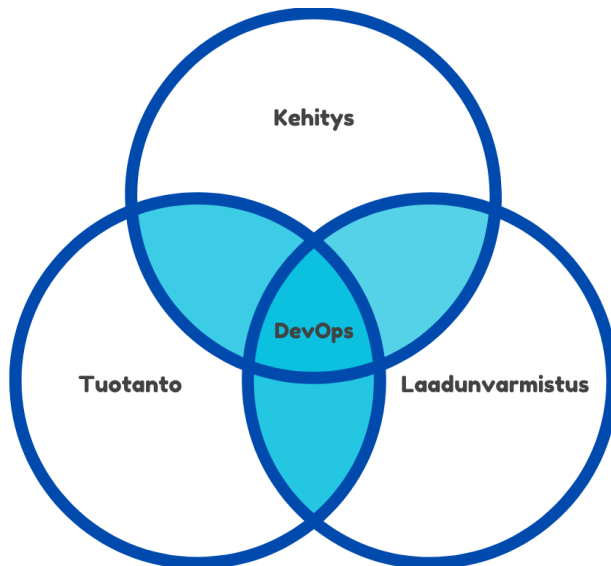
ansaitseen sekä sertifikaatin ja opintoni edistyvät. Olen jo aikaisemmin opinnoissani suorittanut Courseran kautta Meta Back-End Developer sertifikaatin, jonka koin erittäin laadukkaaksi, laajaksi ja hyvin järjestetyksi kokonaisuudeksi. Suorittamani sertifikaatti oli työelämälähtöinen ja sen ansiosta hahmotin ensimmäistä kertaa kokonaiskuvaa työskentelystä sovelluskehittäjänä. Halusin vahvistaa osaamistani back-end puolella ja mielestäni sertifikaatti on hyvä keino näyttää osaamista sen aihealueen piiristä. Sertifikaatti voi auttaa työ- ja harjoittelupaikan löytämisessä, varsinkin jos hakija ei omaa työkokemusta alalta. Se on opintojen lisäksi konkreettinen näyttö hankitusta osaamisesta ja koen sen auttaneen minua harjoittelupaikan löytämisessä.

Ammatillisessa mielessä olen sovelluskehittäjäurani alussa. Kuvailisin itseäni junior full-stack kehittäjäksi, jonka vahvuudet sijaitsevat back-end puolella. Vaikka olen saanut sovelluskehityksen perusteet opinnoistani, pelkkä teoreettinen pohja ei valmista työntekoon. Suoritettu tutkinto antaa minulle kuitenkin valmiudet päästä kehittämään työtä tekemällä. Valitsin tämän sertifikaatin harjoitteluni pohjalta, toivon saavani syvällisemmän ymmärryksen DevOps kehitysmallista ja sen kokonaiskuvasta alkanutta uraani varten, koska jatkan harjoittelun jälkeen työsuhteessa edellä mainitussa yrityksessä. Tulen alkuunsa painottamaan back-end puoleen, mutta tulevaisuudessa tahdon voida työskennellä vapaammin siellä missä tarvitaan.

4 DEVOPS

4.1 Mitä DevOps on?

Suurin osa sovelluskehittäjistä on kuullut termin DevOps, mutta kysyttäessä sen tarkoitusta, saat todennäköisesti aina vähän erilaisen vastauksen. DevOps pitää sisällään kaikki ne toiminnot, jotka parantavat ohjelmistokehityksen laatua, nopeutta, varmuutta sekä yhteisön kommunikaatiota koko ohjelmiston elinkaaren ajan (Valagroup 2020). Sen pohjana on ketterä kehitysmalli ja toimiakseen se vaatii uuden työkuulttuurin omaksumisen, missä kaikki työskentelevät yhteistyössä keskenään jakaen saman maalin ja vastuun (Gitlab 2023). Yksinkertaistettuna DevOps tuo yhteen tuotannon, kehityksen ja laadunvarmistuksen, mikä on havainnollistettu kuvassa 1.



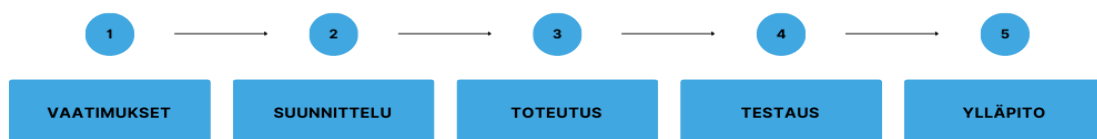
Kuva 1. Yksinkertainen kuvaus DevOpsista.

DevOpsin idean yksinkertaistaminen on tehokas keino havainnollistaa sen kokonaiskuvaa. Rajojen rikkominen tiimien, yksiköiden ja roolien suhteen on vain yksi osa sen kokonaisuudesta. Yläkäsitteenä se on filosofia, joka pyrkii parantamaan ja tehostamaan sovelluskehitystä prosessina alusta loppuun. Se pitää sisällään niin teknisen kuin kulttuurisen puolen. Tekninen puoli nojaa vahvasti automaatioon, mittaamiseen ja seurantaan erilaisten työkalujen avulla. Kulttuurinen puoli painottaa avoimeen kommunikaatioon ja ketterän kehityksen malliin. (Valagroup 2020.)

4.2 DevOpsin tausta

DevOpsin syntyyn vaikuttivat monet merkittävät tekijät, jotka toivat esiin tarpeen kehittää uusia toimintatapoja niin kehityksen kuin tuotannon puolella. Perinteisten sovelluskehitysmallien ristiriidat, kuten erillään toimivat kehitys- ja tuotantotiimit, sekä kasvava monimutkaisuus aina muuttuvassa sovelluskehitystyössä johtivat siihen, että perinteiset kehitysmenetelmät eivät enää pystyneet vastaamaan uusiin haasteisiin. Tämä hidasti toimitusta ja aiheutti ongelmia tiimien yhteistyössä ja kommunikaatiossa. Samaan aikaan automaation ja työkalujen kehittyminen, sekä ketterä kehitysmalli, mahdollistivat joustavamman, toistettavamman ja skaalautuvamman lähestymistavan sovelluskehitykseen. Ketterä kehitys korosti nopeaa palautetta, jatkuvaa kehitystä ja tiivistä yhteistyötä kehitystiimien ja asiakkaiden välillä.

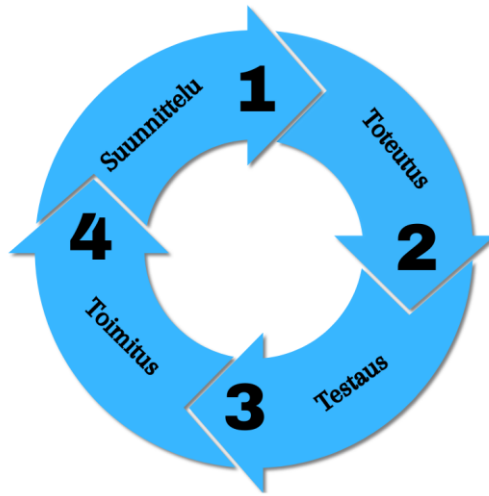
Perinteisistä sovelluskehitysmalleista käytetyin ja tunnetuin on vesiputousmalli (IPM 2022). Vesiputousmalli, joka on esitetty kuvassa 2, on lineaarinen prosessi, jossa jokaisen vaiheen on päätyttävä ennen siirtymistä seuraavaan vaiheeseen. Vesiputousmallin ongelmat sovelluskehityksessä liittyvät sen joustamattomuuteen, hitaaseen toimitukseen ja vähäiseen yhteistyöhön ja kommunikaatioon eri tiimien kesken (Thinking Portfolio 2016). Malli edellyttää tarkat määrittelyt vaatimuksille alkuvaiheessa ja muutosten tekeminen ensimmäisen vaiheen jälkeen hidastaa koko prosessin etenemistä. Eri tiimit työskentelevät peräkkäin ja vuorovaikutus tiimien välillä jaa vähäiseksi. Tuotteen toimitus tapahtuu vasta projektin lopussa, kun kaikki vaiheet ovat suoritettu, jolloin toimitusajat saattavat kestää vuosia. Vesiputousmallia käytetään edelleen projektien kanssa, joissa vaatimukset ovat selvät etukäteen ja tuotteen on pakko olla valmis ensimmäisellä yrittämällä (Boersma 2022).



Kuva 2. Vesiputousmallin elinkaari.

Vuosituhanne vaihteessa, Internetin suosion noustessa, pienet startup-yritykset kyseenalaistivat vesiputousmallin toimivuuden. Vuonna 2001 julkaistu Ketterän ohjelmistokehityksen julistus loi pohjan ketterälle kehitysmallille. Ketterä kehitysmalli on iteratiivinen prosessi, jossa riskit minimoidaan kehittämällä sovelluksia pienissä osissa nopealla vauhdilla. Kuva 3 kuvaa ketterän kehitysmallin. Jokainen iteraatio sisältää samat asiat kuin vesiputousmalli, mutta kestoltaan se on

tyypillisesti 1–2 viikkoa. Huomattavasti lyhyempi kesto mahdollistaa nopean reagoinnin muutoksiin ja jokaisen iteraation jälkeen, projekti arvioidaan uudelleen, jonka jälkeen suunnitellaan seuraava iteraatio. (Hamilton 2023.)



Kuva 3. Ketterän kehitysmallin elinkaari.

DevOps kehitysmalli on kehittynyt ketterän kehitysmallin pohjalle. Vaikka ne eivät ole sama asia, ne voivat täydentää toisiaan. Ketterä kehitysmalli tarjoaa joustavuutta ja nopeaa palautetta kehitystyöhön, kun taas DevOps kehitysmalli mahdollistaa tehokkaan toimituksen ja operoinnin sovelluksen elinkaaren aikana. Monet yritykset yhdistävät nämä kaksi lähestymistapaa luodakseen tehokkaan ja nopean kehitysprosessin. (Vaniukov 2021.)

4.3 DevOpsin käyttö

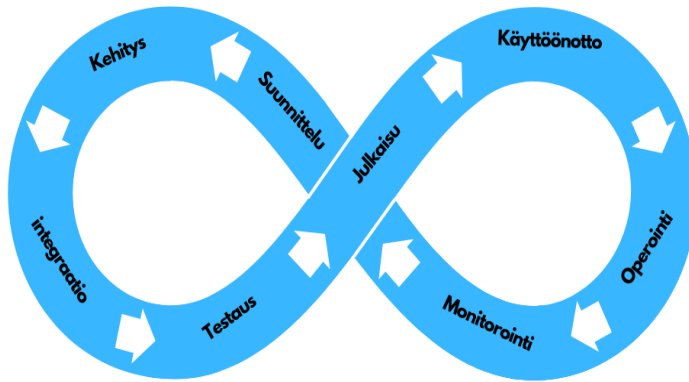
DevOps kehitysmallin käyttöönotto on monimutkainen prosessi, johon kuluu paljon aikaa ja resursseja. Kokonaisvaltainen muutos työskulttuurissa ja uusien mallien sekä työkalujen käyttöönotto on vaikea haaste organisaatioille. Suuri työskulttuurin muutos haastaa niin esihenkilöiden kuin alaisten kyvykkyyden, joka voi johtaa muutosvastarintaan henkilöstön toimesta, vaadituista taidoista voi olla pulaa ja toteutus voi olla epäselvää. Nykyinen infrastruktuuri voi aiheuttaa rajoituksia niin käytettyjen työkalujen kuin vanhojen työmallien suhteen. Käyttöönotto kannattaa tehdä hallitusti ja pienemmissä osissa ja lähestyä haasteita keskittyen DevOps kehitysmallin tarkoitukseen ja siitä koituvaan hyötyyn. Kärsivällisyys ja suunnitelmallisuus ovat olennainen osa onnistunutta käyttöönottoa, ja haasteiden selvittäminen juuritasolla tekee muutosprosessista sujuvampaa niin organisaation kuin henkilöstön näkökulmasta. (KnowledgeHut 2023.)

DevOps kehitysmalli on kokonaisvaltainen tapa tehdä sovelluskehitystä. Se vaatii toimiakseen muutoksia koko organisaatiossa. Työkulttuurin on pohjattava jatkuvaan kommunikaatioon, yhteistyöhön ja jaettuun vastuuseen tiimien välillä. Teknisellä tasolla se vaatii sitoutumista työkaluihin ja automaatioon, jotka pitävät projektit liikkeellä työnkulkujen välillä (IBM 2023). DevOps kehitysmallin käyttöönotossa ja toteuttamisessa seurataan yleensä parhaita käytäntöjä, jotka ovat DevOps yhteisön yleisesti hyväksymiä ja vakiintuneita käytäntöjä kehitysmallin implementointiin. Tärkeimpiin käytäntöihin kuuluu yhteistyöhön perustuvan työkulttuurin luominen, kärsivällisyys, tavoitteiden asettaminen ja niiden seuraaminen, oikeiden työkalujen käyttö, pitkän aikavälin tavoitteet, automaation käyttöönotto ja tietoturvan ja testauksen implementointi koko elinkaaren ajalle (Mohan 2022). Parhaat käytännöt kehittyvät ja muuttuvat yleisen kehityksen myötä, eivätkä ne ole definitiivisiä sääntöjä.

Parhaiden käytäntöjen ohella on tunnistettu useita käytäntöjä, joiden vaikutus DevOps kehitysmallin implementaation on negatiivinen. Näitä kutsutaan englanninkielisellä termillä anti-patterns. Yleisimpiä kompastuskiviä ovat erillisen DevOps tiimin luominen, tiedon ja taidon keskittyminen yksilöihin tiimien sijaan, vauhtisokeus automaation ja muutosten kanssa, uusien työkalujen käyttöönotto ilman oikeaa syytä, jatkuvan parantamisen unohtaminen käyttöönoton jälkeen ja dokumentoinnin sekä tiedonjaon laiminlyönti. (Moustakis 2023.)

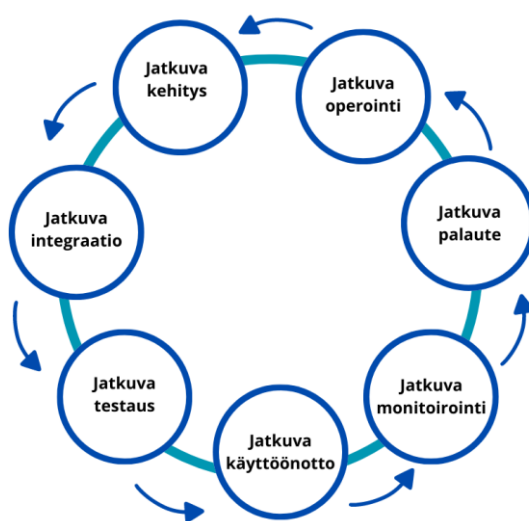
4.4 DevOps kehitysmalli

DevOps kehitysmallin on jatkuva sarja pienempiä työnkuluja, jotka suoritetaan osana isompaa iteratiivista ja automatisoitua kehitysprosessia (IBM 2023). Sen on tapa hallita sovelluksen koko elinkaarta kehityksestä julkaisuun, keskittyen kehityksen ja tuotannon yhteistyöhön. DevOps-malli tukee innovatiivista ajattelua, nopeutta ja kykyä sopeutua muutoksiin sovelluskehitystyössä. DevOps kehitysmallia hyödyntääkseen, tiimillä pitää olla selkeä käsitys mallin eri vaiheista. (Jain 2023.) Kuvassa 3 malli on kuvattu jatkumona kahdeksan eri työnkulun mukaan.



Kuva 4. DevOps kehitysmalli.

Suunnitteluvaiheessa tiimi suunnittelee mitä tehdään iteraation aikana. Kehitysvaiheessa sovelluskehittäjät koodaavat tarvittavat muutokset edeltävän vaiheen mukaan ja valmistavat koodin seuraavaan vaiheeseen. Integraatiovaiheessa edeltävän vaiheen koodi yhdistetään jo olemassa olevaan vanhaan koodiin käyttäen apuna versionhallintaa. Testausvaiheessa tiimi testaa tuotteen integraation jälkeen varmistaen, että se on valmis seuraavaan vaiheeseen. Julkaisuvaiheessa varmistetaan, että tuote läpäisee kaikki sille asetetut vaatimukset eikä esteitä tuotantoon viemiselle ole. Käyttöönottovaiheessa tuote viedään tuotantoympäristöön ja varmistetaan, että tuote toimii suunnitellusti. Operointivaiheessa tuotetta testataan tuotantoympäristössä ja loppukäyttäjät pääsevät käyttämään sitä. Monitorointivaiheessa tuotteen käyttöä, palautetta ja mahdollisia kehityskohteita tunnistetaan ja dokumentoidaan. Seuraavan iteraation suunnittelu pohjaa hyvin vahvasti tähän vaiheeseen. (Atlassian 2023.)



Kuva 5. DevOps kehitysmallin elinkaari.

DevOps kehitysmallin elinkaari voidaan esittää seitsemän jatkumon kautta. Elinkaari on esitetty kuvassa 4. Jatkuva kehitys määrittää koko prosessin suuntaa. Se kattaa kehitysmallista suunnittelu ja kehitysvaiheet. Jatkuva integraatio kattaa kehityksen, integraation ja testauksen. Sen aikana uusi koodi integroidaan olemassa olevaan koodiin ja automatisoidulla testauksella varmistetaan koodin toimivuus. Jatkuva testaus hoidetaan yleensä ennen integraatiota tai sen jälkeen. Jatkuva käyttöönotto kattaa kehitysmallista julkaisun ja käyttöönoton. Jatkuva monitorointi kattaa tuotteen monitoroinnin julkaisun jälkeen. Jatkuvan palautteen aikana asiakkaiden toimintaa seurataan jokaisen julkaisun jälkeen ja heiltä kerätään palautetta. Jatkuva operointi tähtää tuotteen sujuvaan toimintaan ja minimaaliseen häiriöaikaan tuotteen käytössä. (Das 2023.)

4.5 DevOps työkalut

DevOpsin työkalut ja kehitysmalli pohjaavat hyvin vahvasti työkaluihin, jotka tukevat jatkuvaa epäsynkronista yhteistyötä, DevOpsin työnkulkujen integraatiota ja automaatiota. Työkaluista muodostetaan ketju, joka kattaa koko DevOps elinkaaren (Kuva 5). Työkalujen valinta on organisaatioilla, mutta on tärkeää, että ne ovat yhteensopivia mahdollistaakseen häiriöttömän työnkulun. Häiriötön työnkulku on perusta DevOps kehitysmallin toimivuudelle (Atlassian 2023).

Projektinhallinnassa käytetään yleensä ketterää kehitysmallia. Tiimi purkaa projektin vaatimukset pienemmiksi tehtäviksi, joiden etenemistä seurataan viitekehyksen avulla. Viitekehyksen tärkein tehtävä on tuoda läpinäkyvyyttä ja selkeyttä työnjakoon, sekä sen seurantaan. Suosittuja viitekehyksiä projektinhallintaan ovat esim. Scrum, Kanban ja Jira (IBM 2023).

Sovelluskehitystyössä tärkeimmät työkalut liittyvät versionhallintaan ja sen päällä toimivaan CI/CD-putkeen. Versionhallintajärjestelmä rakentuu repositorioiden ympärille. Etärepositorio sisältää lähdekoodin tuoreimman toimivan version. Muutoksia ei tehdä suoraan etärepositorioon, vaan kehittäjä kloonaa siitä itselleen paikallisen version, tekee muutokset siihen ja työntää paikallisen version takaisin etärepositorioon. Sen jälkeen muutokset arvioidaan ja yhdistetään lähdekoodiin, jolloin ne ovat kaikkien projektin kehittäjien saatavilla. Tämä mahdollistaa useamman kehittäjän työskentelyn saman projektin parissa yhtä aikaa. Käytetyimpiä repositorioita ovat GitHub ja GitLab (Ohjelmistokehityksen menetelmät 2021). CI/CD-putki automatisoi jatkuvan integraation ja jatkuvan käyttöönoton. Kun muutokset työnnetään paikallisesta repositoriosta etärepositorioon, se kulkee CI/CD-putken läpi. Jatkuva integraatio varmistaa muutosten toimivuuden automatisoiduilla

testeillä ja ajamalla sovelluksen. Kun muutokset on hyväksytty ja ne yhdistetään lähdekoodiin, jatkuva käyttöönotto julkaisee uuden version sovelluksesta tuotantoympäristöön. (IBM 2023.)

Tuotannon puolella tärkeimmät työkalut liittyvät kokoonpanon hallintaan, jatkuvaan monitorointiin ja jatkuvaan palautteeseen. Kokoonpanon hallinnassa käytetään IaC työkaluja, mitkä mahdollistavat määriteltyjen ja dokumentoitujen ympäristöjen luomisen skriptien avulla. Käytetyimpiä IaC työkaluja ovat esim. Ansible, Chef ja Terraform. Monitoroinnin työkaluja käytetään tunnistamaan ja ratkaisemaan systeemin ongelmia, lisäksi ne keräävät ja analysoivat dataa, mistä näkee kuinka muutokset vaikuttavat sovelluksen toimintaan. Käytetyimpiä monitorointityökaluja ovat esim. Prometheus ja Splunk. Jatkuva palautteessa työkaluja käytetään sovelluksen käytön tallentamiseen, kyselyihin tai suoran palautteen mahdollistamiseen. (IBM 2023, Bose 2023.)

5 PÄIVÄKIRJA

Aloitan sertifiikaatin suorittamisen tutustumalla kokonaisuuteen kuuluviin kursseihin ja niiden rakenteisiin. Teen suunnitelman, että suoritan sertifiikaattiin vaaditut kurssit siinä järjestyksessä, missä on esitetty Courserassa, koska osa loppupään kursseista vaatii aikaisempien kurssien suorittamista pohjaksi. Jokainen kurssi on jaettu kahdesta seitsemään pienempään osaan. Suunnittelen tekeväni pienemmät osat kerralla valmiiksi, jonka jälkeen kirjoitan päiväkirjaan merkinnän kyseisestä osasta. Kun kaikki kurssin osat on tehty, kirjoitan yhteenvedon kurssista.

Kurssien järjestys Courseran alustalla on seuraava:

- Kurssi 1: Introduction to DevOps.
- Kurssi 2: Introduction to Cloud Computing.
- Kurssi 3: Introduction to Agile Development and Scrum.
- Kurssi 4: Introduction to Software Engineering.
- Kurssi 5: Hands-on Introduction to Linux Commands and Shell Scripting.
- Kurssi 6: Getting Started with Git and GitHub.
- Kurssi 7: Python for Data Science, AI & Development.
- Kurssi 8: Python Project for AI & Application Development.
- Kurssi 9: Introduction to Containers w/ Docker, Kubernetes & OpenShift.
- Kurssi 10: Application Development using Microservices and Serverless.
- Kurssi 11: Introduction to Test Driven Development (TDD).
- Kurssi 12: Continuous Integration and Continuous Delivery (CI/CD).
- Kurssi 13: Application Security and Monitoring.
- Kurssi 14: DevOps Capstone Project.

5.1 Kurssi 1: Introduction to DevOps

Osa 1.

Kirjauduin Courseran sivuille ja avasin ensimmäisen kurssin ensimmäisen osan, joka esittelee DevOpsin käsitettä, sekä maalaa kokonaiskuvaa DevOpsista. Tämä osa koostuu suurimmilta osin lyhyistä videotallenteista. Osan puolivälissä ja lopussa on monivalintaharjoitukset, ja viimeinen tehtävä on monivalintakoe. Videoluennot ovat laadukkaita ja niissä käydään läpi DevOpsin

määritelmää, kuvausta ja historiaa. Monivalintakokeessa on 10 kysymystä ja läpäisemiseen vaaditaan vähintään 7 oikein. Olin tyytyväinen läpäistessäni monivalintakokeen ensimmäisellä kerralla saaden 9 oikein.

Osa 2.

Toinen osa esittelee sovelluskehityksen sosiaalista puolta, DevOps mallin hyötyjä ja epäonnistumisten kääntämistä hyödyksi. Osa koostuu videoluennoista ja kahdesta monivalintaharjoituksesta ja viimeisenä tehtävänä on taas monivalintakoe. Pidin videoluennoista, ne ovat lyhyitä ja ytimekkäitä. Monivalintakoe meni taas ensimmäisellä yrittämällä läpi.

Osa 3.

Viikon tauon jälkeen aloitin kolmannen osan tekemisen. Osa esittelee DevOpsin periaatteita ja tekniikoita. Osa koostuu lyhyistä videoluennoista, kahdesta monivalintaharjoituksesta ja monivalintakokeesta. Lisäksi siinä esitellään mielenkiintoinen esimerkki työelämästä, missä DevOpsin mukainen kehitysmalli olisi voinut estää Knight Capital yrityksen katastrofaalisen virheen ja konkurssin.

Osa 4.

Tässä osassa käsitellään DevOpsia organisaation näkökulmasta, sekä sitä miten se implementoidaan ja miten sen toimivuutta voidaan seurata. Osa koostuu videoluennoista, monivalintaharjoituksista sekä monivalintakokeesta. Lisänä tässä osassa on kaksi harjoitusta. Toinen koski DevOpsin toimintaa ja toisessa valittiin oikeita ratkaisuja seurantaan liittyen.

Osa 5.

Viimeisessä osassa esitellään kolme tapaustutkimusta DevOps kehitysmalliin, organisaatioon ja sosiaaliseen sovelluskehitykseen liittyen. Osa sisältää myös monivalintakokeen kattaen koko kurssin. Ensimmäisellä yrittämällä monivalintakoe epäonnistui, toisella yrittämällä kertasin kaikki kurssin yhteenvedot ja läpäisin sen. Seuraavia kursseja silmällä pitäen, tiedän viimeisen kokeen kattavan koko kurssin.

Kurssin yhteenveto.

Sain ensimmäisen kurssin tehtyä viidessä illassa ja aikaa kului noin 10 tuntia. Kurssi koostuu lähinnä videoluennoista ja monivalintatehtävistä. Videoluennot ovat laadukkaita ja lyhyitä tiivistäen teorian selkeästi ja kattavasti. Kurssin tarkoitus on esitellä DevOps kokonaisuutena, ja mielestäni

se onnistui hyvin perusteellisesti. Etenkin viimeisessä osassa tehdyt tapaustutkimukset jäivät positiivisesti mieleen. Niissä rakennettiin DevOps suunnitelmat kolmelle erilaiselle yritykselle ja se antoi käytännön näkökulmaa DevOpsin implementaatioon niin kehittäjän kuin yrityksen näkökulmasta. Kurssi sisältää pituuteensa nähden valtavasti tietoa, mutta siinä ei menty vielä pintaa syvemmälle tekniikoiden tai työkalujen kanssa.

Kurssi laajensi ymmärrystäni DevOpsista ja mielikuvani siitä kehitysmallina muuttui. DevOps on paljon enemmän kuin uusi tapa tehdä sovelluskehitystä. Toimiakseen se tarvitsee työkalutuurin muutoksen organisaatiossa. Yhteistyö tiimien välillä, jaettu vastuu, läpinäkyvyys työnteossa ja yhteinen päämäärä on pohja DevOpsin toimivuudelle. Käyttäkseen DevOpsin tekniikoita ja työkaluja, pohjan on oltava kunnossa, muuten hyöty jää heikoksi. DevOpsin käyttöönotto organisaatiossa on myös iso ja monimutkainen prosessi, joka vaatii hyvän ja selkeän suunnitelman, sekä henkilöstön kouluttamista.

5.2 Kurssi 2: Introduction to Cloud Computing

Osa 1.

Aloitin uuden kurssin, jonka aiheena ovat pilvipalvelut. Ensimmäinen osa selittää mitä käsite tarkoittaa ja pitää sisällään. Osa koostuu videotallenteista, yhteenvedosta ja monivalintakokeesta. Mukana on myös tapaustutkimuksia pilvipalveluiden implementaatiosta neljässä yrityksessä, tätä lisää pidin mielenkiintoisina ja avaavana lisänä.

Osa 2.

Toisessa osassa esitellään erilaisia pilvipalveluita ja ympäristöjä. Se koostuu videoluennoista, yhteenvedosta, labrasta ja monivalintakokeesta. Labrassa tein IBM Cloud tunnukset kurssin tarjoamalla opetuskoodilla ja unohduin tutkiskelemaan sitä ympäristöä tunniksi. Tunnuksia tulen tarvitsemaan kurssin viimeisessä tehtävässä, joka palautetaan ja vertaisarvioidaan.

Osa 3.

Kolmannessa osassa esitellään pilvipalveluiden arkkitehtuuria ja erilaisia tallennusmalleja pilvessä. Osa koostuu videoluennoista, yhteenvedosta ja monivalintakokeesta. Videoluennot ovat

mielenkiintoisia ja laadukkaasti tuotettuja, mutta olisi virkistävää, jos niiden välissä olisi esimerkiksi monivalintaharjoituksia. Uutta tietoa tulee paljon ja monivalintakoe on haastava. Lämpäisen monivalintakokeen vasta toisella yrittämällä, kun olen kerrannut materiaalin videotallenteiden transkriptioista.

Osa 4.

Neljännessä osiossa tutustun pilvipalveluiden trendeihin sekä DevOpsin rooliin pilvipalveluiden monimutkaisuusien hallinnassa. Osa koostuu videoluennoista, yhteenvedosta ja monivalintakokeesta. Edeltävä osio tuntui hieman puuduttavalta, mutta en huomannut samaa ongelmaa tämän osion suhteen, vaikka rakenne on samanlainen. Monivalintakoe menee läpi täysin pistein ensimmäisellä yrittämällä, mihin olen tyytyväinen.

Osa 5.

Viimeinen osa sisältää koko kurssin kattavan monivalintakokeen ja vertaisarvioidun tehtävän. Kertaan kurssin sisältämät yhteenvedot ja lämpäisen monivalintakokeen ensimmäisellä yrittämällä. Vertaisarvioidussa tehtävässä luodaan instanssi Cloud Object Storagesta IBM Cloud palvelussa. Tehtävä dokumentoitiin kuvankaappauksin ja ne palautettiin kurssialustalle instanssin julkisen www-osoitteen kanssa.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana käyttäen siihen noin 15 tuntia. Kurssi koostuu suurimmilta osin videoluennoista. Kurssin videoluennot ovat laadukkaita ja mielenkiintoisia mutta huomaan kertaavani ne lukemalla transkription, en uudelleenkatsoamalla niitä. Kurssin kohokohta on IBM Cloud palvelun käyttö, koska tutustuin ensimmäistä kertaa pilvipalveluympäristöön. Viimeinen vertaisarvioitu tehtävä oli hankala monimutkaisten ja sekavien ohjeiden takia, mutta sen lisäksi en keksi kurssista huonoa sanottavaa. Kurssin aihe on laaja, mutta teoria on esitetty hyvin tiivistettynä ja kattavasti.

Pilvipalvelut liittyvät DevOpsiin hyvin olennaisesti ja nykyään suurin osa sovelluksista pyörii pilvipalveluiden päällä. Niiden käyttö tehostaa kehitysprosesseja, nopeuttaa toimitusta ja parantaa sovellusten laadunvalvontaa. Pilvipalveluiden tärkeimpiä ominaisuuksia DevOpsin suhteen ovat joustavat infrastruktuuripalvelut ja työkalut niiden automatisointiin ja hallintaan. Esimerkiksi CI/CD-putki rakentuu pilvipalveluiden päälle ja se on olennainen osa DevOps kehitysmallia.

5.3 Kurssi 3: Introduction to Agile Development and Scrum

Osa 1.

Aloitan uuden kurssin, joka käsittelee ketteriä menetelmiä. Ensimmäinen osa esittelee ketterää ideologiaa ja sen käyttöä Scrum viitekehysessä. Osa koostuu videoluennoista, monivalintaharjoituksista, yhteenvedoista ja monivalintakokeesta. Koen osan syventävän ymmärrystäni ketteristä menetelmistä. Minulla on jo kokemusta Scrumin käytöstä kouluprojekteissa ja työharjoittelussa. Monivalintakoe menee läpi täysin pistein ensimmäisellä yrittämällä.

Osa 2.

Toinen osa keskittyy iteratiiviseen suunnitteluun. Osa koostuu videoluennoista, monivalintaharjoituksista, yhteenvedoista ja monivalintakokeesta. Lisäksi osa sisältää labroja, joissa harjoitellaan Kanban taulun käyttöä ZenHub työkalulla. Labrat tuovat mukavaa vaihtelua opiskeluun.

Osa 3.

Viimeinen osa esittelee miten Scrum malli implementoidaan organisaatiossa ja miten sitä käytetään. Osa koostuu videoluennoista, monivalintaharjoituksista, yhteenvedoista, labroista ja koko kurssin kattavasta monivalintakokeesta. Monivalintakoe menee läpi ensimmäisellä yrittämällä ja labrat olivat hyvä lisä kurssin toteutukseen.

Kurssin yhteenveto.

Suoritin kurssin yhden päivän aikana käyttäen siihen noin 10 tuntia. Kurssin toteutus oli tehty laadukkaasti ja kattavasti. Kurssin kohokohta oli labrat, missä ZenHubia käyttäen toteutettiin kokonainen iteraatio. Pidin kurssin rakennetta onnistuneena, ensiksi esiteltiin teoriaa videoluentoina ja sen jälkeen se toteutettiin labrassa. Kurssin aihe oli hyvin rajattu ja se eteni johdonmukaisesti. En suunnitellut tekeväni kurssia yhdeltä istumalta, en vain malttanut lopettaa aloitettuani. Kurssi syvensi ymmärrystäni ketteristä menetelmistä ja siitä mikä rooli, sillä on DevOps kehitysmallissa.

DevOps on kehittynyt ketterien menetelmien pohjalle. Ne eivät ole sama asia, vaan ne täydentävät toisiaan. Ketterä kehitysmalli tuo joustavuutta ja nopeaa palautetta kehitystyöhön, kun taas DevOps kehitysmalli mahdollistaa tehokkaan toimituksen ja operoinnin sovelluksen elinkaaren aikana.

Monesti organisaatiot yhdistävät nämä kaksi mallia. Ketterä kehitysmalli voidaan nähdä yhtenä tärkeimpänä DevOps kehitysmallin projektinhallinnan työkaluna.

5.4 Kurssi 4: Introduction to Software Engineering

Osa 1.

Aloitetaan uuden kurssin, joka käsittelee sovelluskehitystä kokonaisuutena. Ensimmäinen osa esittelee sovelluskehityksen työnkuvaa ja sovelluskehityksen elinkaaren. Osa koostuu videoluennosta, monivalintaharjoituksista ja monivalintakokeesta. Lisänä yksi tehtävä missä tunnustetaan sovelluskehityksen elinkaaren eri vaiheet.

Osa 2.

Toinen osa esittelee front-end ja back-end sovelluskehityksen eroja ja niiden käyttämiä tekniikoita ja työkaluja. Osa koostuu videoluennosta, monivalintaharjoituksista, labrasta ja monivalintakokeesta. Labrassa tutustutaan kurssin koodaustehtävissä käytettävään virtuaaliseen IDE-ympäristöön, koodauskielenä käytetään pythonia. IDE toimii yllättävän hyvin virtuaalisesti, valmiin python ympäristön saaminen napinpainalluksella tulee säästämään aikaa ja hermoja.

Osa 3.

Kolmas osa esittelee ohjelmoinnin logiikkaa, suunnittelua ja koodin organisointia. Osa koostuu videoluennosta, monivalintaharjoituksista, labroista ja monivalintakokeesta. Labroissa harjoitellaan koodauksen alkeita python kielellä. Labrat on sijoitettu aina teorian jälkeen, jolloin opittua pääsee konkreettisesti kokeilemaan. Tämä lisää mielestäni opiskelun mielekkyyttä.

Osa 4.

Neljäs osa esittelee sovelluskehityksen arkkitehtuuria ja suunnittelua. Osa koostuu videoluennosta, monivalintaharjoituksista, yhteenvedosta ja monivalintakokeesta. Lisäksi yhdessä tehtävässä luotiin arkkitehtuurinen diagrammi. Pidin erityisesti sovelluskehityksen arkkitehtuuriin tutustumisesta, se avartaa kokonaiskuvaa sovelluskehityksestä.

Osa 5.

Viidennessä osassa pohditaan sovelluskehittäjien erilaisia työnkuvia ja urapolkuja sekä sovelluskehityksen eettisiä sääntöjä. Osa koostuu videoluennosta, monivalintaharjoituksesta,

yhteenvedosta ja monivalintakokeesta. Pidin erityisesti videoluennosta missä työntekijät omin sanoin kuvasivat heidän työtänsä. Ne antoivat hyviä esimerkkejä erilaisista työnkuvista sovelluskehityksessä.

Osa 6.

Viimeinen osa sisältää sanaston koko kurssista ja koko kurssin kattavan monivalintakokeen. Kertaan osien yhteenvedot sekä sanaston ja läpäisen monivalintakokeen ensimmäisellä yrittämällä. Pidin siitä, että kurssiin oli lisätty sanasto, tallensin siitä kopion omaan käyttöön.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana käyttäen siihen noin 20 tuntia. Kurssin toteutus on laadukas ja aihe erittäin laaja. Videoluennot ovat lyhyitä ja kuvaavat teorian tiiviisti ja kattavasti, sekä labrat ovat hyvin ohjeistettuja. Laaja aihe on tiivistetty hyvin ymmärrettäväksi kokonaisuudeksi ja sovelluskehitys kuvataan kattavasti ruohonjuuritasolta lähtien. Juuri tämä lähestymiskeino auttaa hahmottamaan sovelluskehityksen kokonaiskuvaa. Pidin kurssista hyvin paljon koska se avasi sovelluskehitystä kokonaisuutena kattavasti ja antoi erilaisia esimerkkejä työrooleista sekä niiden työnkuvista. Kurssi syvensi ymmärrystäni sovelluskehityksen elinkaaresta sekä suunnittelusta ja arkkitehtuurista sovelluskehityksessä. Sovelluskehitysprosessin ymmärtäminen kokonaisuutena on tärkeä osa DevOps kehitysmallin implementaatiota, koska yleisimmät vastoinkäymiset johtuvat henkilöstön koulutuksen puutteesta ja siitä johtuvasta epäselvyydestä prosessissa. Kurssissa käytettiin ensimmäistä kertaa Coursera alustan virtuaaliympäristöä koodaamiseen. Tiesin kurssin sisältävän pythonia, ja yllätyin positiivisesti, kun huomasin kurssin olevan suoritettavissa pelkästään selaimessa. Kurssi sujui mallikkaasti, enkä kohdannut ongelmia sitä suorittaessani.

5.5 Kurssi 5: Hands-on Introduction to Linux Commands and Shell Scripting

Osa 1.

Aloitan uuden kurssin, joka käsittelee Linux komentoja ja Bash shell skriptausta. Ensimmäinen osa esittelee Linuxin perusteita, taustaa ja ominaisuuksia. Osa koostuu videoluennosta, labrasta, yhteenvedosta ja monivalintakokeesta. Videoluennot pohjustavat labraa, missä harjoitellaan Linuxin terminaalien käyttöä. Videotallenteet ovat lyhyitä ja ytimekkäitä ja labra on toteutettu SN Labsin virtuaaliympäristössä. Pidän paljon mallista missä ensimmäiseksi opiskellaan teoriaa ja sen jälkeen harjoitellaan opittua asiaa käytännön tasolla.

Osa 2.

Toinen osa esittelee laajan kattauksen Linux komentoja. Osa koostuu videotallenteista, kahdesta labrasta, yhteenvedosta ja monivalintakokeesta. Labroissa harjoiteltiin navigointia tiedostojen hierarkiassa ja tiedon arkistointia sekä pakkaamista .zip tiedostoon. Pidin labroista ehdottomasti eniten käyttäen niihin paljon aikaa. Jouduin uusimaan monivalintakokeen kaksi kertaa ja mielestäni se oli haastava.

Osa 3.

Kolmas osa käsittelee shell skriptausta. Osa koostuu videotallenteista, kolmesta labrasta, yhteenvedosta ja monivalintakokeesta. Lisäksi alusta tarjoaa Linux and Bash Command Cheat Sheet kokoelman minkä tallensin omaa käyttöä varten. Videoluennot pohjustavat labroja, ja nautin niiden suorittamisesta. Monivalintakoe oli haastava, mutta läpäisin sen ensimmäisellä yrittämällä.

Osa 4.

Viimeinen osa sisältää koko kurssin kattavan monivalintakokeen ja vertaisarvioitavan projektin. Projektissa tein automatisoidun skriptin, joka loi varmuuskopiot projektin muutetuista tiedostoista 24 tunnin välein. Projekti oli erittäin mielenkiintoinen koska se kokosi hyvin yhteen kurssin aikana oppimani asiat. Projekti dokumentoitiin kuvankaappauksin ja palautettiin arvosteltavaksi Courseran kurssialustalle. Lisävaatimuksena projektin suorittamiseen, minun piti suorittaa vertaisarviointi kahdelle kanssaopiskelijalle.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana ja käytin siihen noin 16 tuntia. Kurssi oli toteutettu hyvin ja sen suorittaminen oli sujuvaa. Videoluennot tiivistivät teorian kattavasti ja labrojen ohjeistus oli selkeää. Videoluennot pohjustivat labroja hyvin ja labrat itsessään olivat hyvä keino harjoitella opittua teoriaa käytännössä. Vertaisarvioitu projekti oli kurssin mielenkiintoisin osuus, koska siinä pääsi konkreettisesti toteuttamaan kurssissa opittua teoriaa. Linux komentojen käyttö on erittäin yleistä sovelluskehityksessä. Niiden käyttö mahdollistaa keskeytymättömän työskentelyn ja ne nopeuttavat sovelluskehitysprosessia kokonaisuutena. Niiden pitäisi kuulua jokaisen sovelluskehittäjän perustaitoihin.

DevOps pohjaa hyvin vahvasta automaation käyttöön toimituksen nopeuttamiseksi. Skriptausta käytetään usein tuotannon ja kehityksen puolella vaiheiden automatisoinnissa. Se on tärkeässä

osassa niin ympäristöjen hallinnassa, konfiguraationhallinnassa, CI/CD-putken automatisoinnissa sekä hälytys ja valvontajärjestelmien hallinnassa. Skriptaus parantaa prosessien tehokkuutta, vähentää inhimillisiä virheitä ja vahvistaa jatkuvaa toimitusta ja laadunvalvontaa.

5.6 Kurssi 6: Getting Started with Git and GitHub

Osa 1.

Aloitan uuden kurssin, joka käsittelee repositorioiden käyttöä sovelluskehityksessä. Kurssissa käytetään versionhallintajärjestelmänä Gittiä ja repositoriona GitHubia. Ensimmäinen osa koostuu videoluennoista, labrasta ja monivalintakokeesta. Siinä esitellään Gitin ja GitHubin perusteita. Labrassa tehtävänä oli luoda tili GitHubiin, mikä minulla on jo valmiina koska olen käyttänyt GitHubia kouluprojektien teossa. Monivalintakoe oli helppo, koska minulla on jo kokemusta Gitin käytöstä niin koulun kuin harjoittelunkin puolesta.

Osa 2.

Toinen osa käsittelee Gitin toimintaa, käskyjä ja työnkulkua. Osa koostuu videoluennoista, kolmesta labrasta ja monivalintakokeesta. Videoluennot ovat lyhyitä ja ytimekkäitä. Labraoissa harjoitellaan haarojen käyttämistä, projektien kloonaamista ja Gitin käskyjä kurssin tarjoamassa virtuaaliympäristössä. Osassa on myös vapaaehtoinen lisätehtävä missä opastetaan Gitin ja GitHubin käyttöönotto omalle tietokoneelle. Lisätehtävä kannattaa tehdä koska kurssin lopputyö on käytännön harjoitus repositorion käytöstä. Siirryn seuraavaan osioon koska olen jo ottanut Gitin ja GitHubin käyttöön aikaisemmin. Monivalintakoe tuntui helpolta ja meni läpi ensimmäisellä yrittämällä.

Osa 3.

Viimeinen osa sisältää vertaisarvioitavan projektin, jonka pohjalta kurssi arvostellaan. Projekti sisälsi kattavasti kurssissa opetetut asiat ja pidin sitä hyvin opettavaisena. Projekti palautettiin arvosteltavaksi Courseran alustalle linkkinä omaan GitHub projektiin. Projektin läpäiseminen vaatii myös kahden vertaisarvioinnin tekemisen kanssaopiskelijoille.

Kurssin yhteenveto.

Suoritin kurssin yhden päivän aikana ja käytin siihen noin 6 tuntia. Kurssi oli lyhyempi entä aikaisemmat kurssit, mutta se johtui aiheesta, joka ei ole hirveän laaja. Gitin ja repositorioiden

käytön perusteet oppii nopeasti. Kurssi sisälsi kattavan kuvauksen Gitin käytöstä ja sen käyttöönotosta sovelluskehityksessä. Kurssissa oli monta labraa, jotka olivat selkeitä ja hyvin suunniteltuja. Vertaisarvioitava projekti toimi hyvänä esimerkkinä Gitin ja repositorion käytöstä sovelluskehityksessä. Vaikka kurssin aihe ei ole hirveän laaja on se sitäkin tärkeämpi. Versionhallintajärjestelmien ja repositorioiden käyttö kuuluu sovelluskehittäjien jokapäiväiseen työkuvaan.

Versionhallintajärjestelmät ja repositoriot ovat olennainen työkalu DevOps kehitysmallissa. Ne mahdollistavat useamman sovelluskehittäjän työskentelyn saman projektin parissa, mahdollistavat CI/CD-putken toiminnan sekä koodin laadun varmistamisen ja mahdollistavat prosessien automatisoinnin. Nämä kaikki tukevat DevOpsin toimintatapoja, joihin kuuluu jatkuva yhteistyö, nopea julkaisu ja korkea laatu.

5.7 Kurssi 7: Python for Data Science, AI & Development

Osa 1.

Aloitin uuden kurssin, joka käsittelee sovelluskehityksen alkeita Python kielellä. Kurssin tarkoitus on toimia pohjana tuleville Python kursseille, joita sertifiikaatti sisältää. Ensimmäinen osa koostuu videoluennoista, kahdesta labrasta ja monivalintakokeesta. Labrat tehdään Coursera alustan tarjoamassa virtuaaliympäristössä. Videoluennot ovat lyhyitä ja ytimekkäitä ja labrat yksinkertaisia sekä hyvin ohjeistettuja. Minulla on kokemusta Python kielellä koodaamisesta ja koen osan olevan yksinkertainen ja helppo. Videoluennot pitää katsoa alusta loppuun, että pääsen jatkamaan osan suoritusta, joten nopeutan niitä.

Osa 2.

Toinen osa käsittelee list, tuple ja set datamalleja. Osa koostuu Videoluennoista, labroista ja monivalintakokeesta. Malliltaan osa on samanlainen kuin ensimmäinen osa. Osan aihe on Pythonin alkeita ja suoritan sen nopeasti ilman ongelmia loppuun.

Osa 3.

Kolmannessa osassa jatketaan Pythonin perusteiden parissa. Osa esittelee objektit, luokat ja funktiot. Osa koostuu videoluennoista, labroista, monivalintaharjoituksista sekä

monivalintakokeesta. Aihe on edelleen Pythonin perusteita ja koen sen yksinkertaiseksi ja helpoksi aikaisemman kokemuksen pohjalta.

Osa 4.

Neljännessä osassa siirrytään datan käsittelyyn Pythonilla. Osa esittelee Pandasin ja Numbyn käyttöä datan muokkaukseen. Osa koostuu videoluennoista, labroista, monivalintaharjoituksista sekä monivalintakokeesta. Videoluennot pohjaavat labroja hyvin, ja pidän labrojen toteutuksesta. Coursera alustan tarjoama virtuaaliympäristö tekee opiskelusta sujuvaa. Labrojen suorittaminen vei yllättävän paljon aikaa, mutta ne olivat hyvin ohjeistettuja, mikä teki niiden suorittamisesta sujuvaa.

Osa 5.

Viimeisessä osassa käsitellään datan keräämistä API:n ja verkkoharavoinnin avulla. Osa koostuu videoluennoista, labroista, monivalintaharjoituksista, sanastosta ja koko kurssin kattavasta monivalintakokeesta. Koko kurssin kattava monivalintakoe oli mielestäni vaikea koska aihe oli laaja ja kysymykset hyvin tarkkoja. Läpäisin monivalintakokeen toisella kerralla, kerrattuani huolellisesti koko kurssin.

Kurssin yhteenveto.

Suoritin kurssin kahden viikon aikana ja käytin siihen noin 25 tuntia. Kurssi oli hyvin toteutettu, videoluennot tiivistivät teorian kattavasti ja labrat olivat hyvin ohjeistettuja. Labroissa käytetty valmis virtuaaliympäristö mahdollisti keskittymisen Pythonilla koodaamiseen, eikä aikaa kulunut Python ympäristön pystyttämiseen omalle koneelle. Kurssi opetti Pythonin alkeet hyvin perusteellisesti ja pidin erityisesti datan muokkausta ja keräämistä koskevista osista. Kurssi on suunniteltu ensimmäiseksi Python kurssiksi opiskelijoille, joilla ei ole kokemusta Pythonista ja se tarjoaa siihen hyvän perustan. Pidin erityisesti virtuaaliympäristön käyttämisestä koska se mahdollistaa keskittymisen olennaiseen ja tekee kurssin suorittamisesta sujuvaa. Pythonia käytetään erityisesti data analyysissä, datan visualisoinnissa ja tehtävien automatisoinnissa, mutta se soveltuu myös kokonaisvaltaiseen sovelluskehitykseen. Pythonia pidetään helpohkona kielenä oppia ja selkolukuisena, mikä on johtanut sen suureen suosioon.

5.8 Kurssi 8: Python Project for AI & Application Development

Osa 1.

Aloitan uuden kurssin, joka on jatkoa edeltävälle kurssille. Tässä kurssissa kehitetään nettisovellus Pythonin Flask viitekehysellä. Edeltävän kurssin hyväksytyt suorittaminen on edellytyksenä tämän kurssin aloittamiseen. Ensimmäinen osa esittelee yksikkötestausta, Python pakettien luomista ja Flask viitekehysten, jota käytetään paljon nettisovellusten kanssa. Osa koostuu videoluennoista, labroista, kirjallisista teoriaosuuksista ja monivalintakokeesta. Videoluennot ovat lyhyitä mutta oleellinen tieto on tiivistetty hyvin. Teoriaosuus selvittää yksikkötestauksen ja Python pakettien tarkoituksen ja labroissa tehdään harjoitusnettisovellus sekä luodaan yksikkötestejä. Monivalintakoe oli haastava vaikeiden kysymysten takia, mutta läpäisin sen ensimmäisellä yrittämällä.

Osa 2.

Toinen ja samalla viimeinen osa sisältää vertaisarvioidun nettisovelluksen teon. Kurssissa tehtävä sovellus on ohjelmointirajapinta, mihin luodaan kaksi pääteipistettä tekstin kääntämiseksi englannista ranskaksi ja ranskasta englanniksi. Osa koostuu kirjallisista teoriaosuuksista, labran valmistelusta, labrasta ja vertaisarvioinnista. Labrassa sovellus koodataan Courseran alustan tarjoamassa virtuaaliympäristössä. Sovelluksen kehitys sujui hyvin laadukkaasti ohjeistuksen ansiosta. Sain työni valmiiksi ilman ongelmia ja pidin sitä mielenkiintoisimpana työnä mitä sertifikaattikokonaisuus on tähän mennessä tarjonnut. Olen harjoittelussani päivittänyt useita rajapintoja ja oli mukava luoda oma rajapinta sekä syventää osaamistani aiheen ympärillä. Sovellus palautettiin arvosteltavaksi Courseran alustalle linkkinä omaan GitHub repositorioon. Kurssin suorittaminen edellytti minulta vielä kahta vertaisarviointia kanssaopiskelijoiden töistä.

Kurssin yhteenveto.

Suoritin kurssin yhden päivän aikana ja käytin siihen noin 10 tuntia. Kurssi oli erittäin opettavainen kokemus nettisovelluksen kehittämisestä. Kurssin aihe oli mukavan rajattu, koska se pohjasi edeltävään Python kurssiin. Kurssimateriaalit olivat yhtä laadukkaasti toteutettu entä aikaisemmissa kurseissa ja suorittaminen kävi sujuvasti. Oman ohjelmointirajapinnan kehittäminen on hyvä tapa näyttää osaamistaan sovelluskehittäjänä. Sertifikaatin suorittamisessa tehtävien projektien lisääminen omaan repositorioon on järkevä ratkaisu. Henkilökohtaisen repositorion osoitteen lisääminen ansioluetteloon on konkreettinen tapa näyttää osaamista työnhakutilanteessa.

Nettisovellukset voivat tarjota monia ominaisuuksia ja tekniikoita, joilla voidaan tukea DevOps kehitysmallia. Nettisovelluksia käytetään työkaluina niin CI/CD-putken osina kuin automaation tukena testauksessa, koodin tarkastamisessa, ympäristöjen hallinnassa ja sovellusten käyttöönotossa. Työkalut mahdollistavat DevOps kehitysmallin elinkaaren eri vaiheet.

5.9 Kurssi 9: Introduction to Containers w/ Docker, Kubernetes & Openshift

Osa 1.

Aloitan uuden kurssin, joka käsittelee konttitekniikan käyttämistä sovelluskehityksessä ja pilvipohjaisia sovelluksia. Ensimmäinen osa koostuu videoluennosta, labrasta, yhteenvedosta ja monivalintakokeesta. En ole aikaisemmin tutustunut konttitekniikkaan ja pidän aihetta erittäin mielenkiintoisena. Videoluennot ovat hyvin tiivistettyjä tietopaketteja ja labra on hyvin ohjeistettu. Labrassa harjoitellaan Dockerin käyttöä. Ennen monivalintakoea kertaan koko osan huolella, koska uutta tietoa on paljon. Monivalintakoe oli haastava koska kysymykset ovat hyvin tarkkoja ja uutta teoriaa on paljon. Läpäisin monivalintakokeen ensimmäisellä yrittämällä, johon olin erittäin tyytyväinen.

Osa 2.

Toinen osa käsittelee konttien hallinnan perusteita Kubernetesin avulla. Osa koostuu videoluennosta, labrasta, yhteenvedosta ja monivalintakokeesta. Videoluennot ovat lyhyitä ja ytimekkäitä ja sisältävät paljon tietoa tiivistetyssä muodossa. Päätän katsoa ne toistamiseen ennen labraan siirtymistä. Labrassa käytetään Courseran alustan virtuaaliympäristöä ja siinä harjoitellaan Kubernetes podien luominen, käyttöönotto, ReplicaSet määrittely ja kuormanjako. Ennen monivalintakoea päätän taas kerrata koko teoria osan koska uutta tietoa on niin paljon. Monivalintakoe tuntui helpolta, vaikka kysymykset olivat hyvin yksityiskohtaisia.

Osa 3.

Kolmas osa käsittelee konttien hallintaa Kubernetesin avulla syvällisemmin. Osa koostuu videoluennosta, labrasta, yhteenvedosta ja monivalintakokeesta. Käytän samaa lähestymistä tämän osan kanssa entä edeltävän. Labrassa harjoitellaan Kubernetes podien automaattista skaalautumista ja kontissa olevien sovellusta päivittämistä. Läpäisen monivalintakokeen vasta toisella yrittämällä, uutta tietoa on paljon ja kysymykset ovat hyvin tarkkoja. Teoriaosuuksien

kertaaminen ei tunnu haittaavan, tieto on esitetty tiiviisti ja siihen ei kulu hirveästi aikaa. Henkilökohtainen mielenkiinto konttitekologiaan saattaa myös vaikuttaa opiskeluuntooni.

Osa 4.

Neljännessä osassa tutustutaan Red Hatin OpenShiftiin ja Kubernetesin ekosysteemiin. Osa koostuu videoluennoista, labrasta, yhteenvedosta ja monivalintakokeesta. Videoluennot ovat yhtä laadukkaita kuin aikaisemmissa osissa ja labrassa harjoitellaan OpenShiftin käyttöä. Osan tekeminen on sujuvaa enkä törmää ongelmiin. Päätän kerrata teoriaosuudet ennen monivalintakoetta ja läpäisen sen ensimmäisellä yrittämällä.

Osa 5.

Viimeinen osa sisältää vertaisarvioitavan projektin missä kehitetään yksinkertainen vieraskirja sovellus, joka julkaistaan Dockerin ja Kubernetesin avulla. Osa koostuu ohjeistuksesta, labrasta ja vertaisarvioinnista. Labrassa käytetään Courseran alustan tarjoamaa virtuaaliympäristöä kehitystyöhön. Projektin läpivienti onnistui hyvin, erinomaisen ohjeistuksen ansiosta. Projekti vietiin henkilökohtaiseen repositorioon ja palautettiin linkkinä Coursera alustalle vertaisarviointiin. Osan suorittaminen vaati minulta vielä kahta vertaisarviointia kanssaopiskelijoiden projekteihin.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana ja käytin siihen noin 20 tuntia. Kurssin arvioitu kesto Courseran mukaan on noin 13 tuntia, mutta aiheen kiinnostavuuden ja tärkeyden takia keskityin siihen huolella. Kurssi oli hyvin suunniteltu kokonaisuus ja antoi laajan ymmärryksen konttitekologian perusteista, taustasta ja käytöstä. Olen törmännyt Dockeriin ja OpenShiftiin jo harjoitteluni aikana ja koen aiheen erittäin tärkeäksi osaksi kokonaisvaltaista sovelluskehitystä. Kurssin kohokohta minulle oli oman sovelluksen kehittäminen ja julkaiseminen. Pidän myös siitä, että kurssit käyttävät henkilökohtaista repositoriota, jolloin valmis projekti jää minulle todisteeksi tehdystä työstä. Kurssi oli erittäin opettavainen ja perusteellinen, suosittelen sitä lämpimästi jokaiselle aiheesta kiinnostuneelle.

Konttitekologian käyttäminen on olennainen osa DevOps kehitysmallia. Se mahdollistaa sovelluksen ja sen riippuvuuksien pakkaamisen yhtenäiseen ympäristöön kontin sisälle varmistaen sovelluksen toimivuuden kaikissa ympäristöissä. Se helpottaa sovellusten käyttöönottoa, skaalautuvuutta, jatkuvaa integraatiota ja jatkuvaa toimitusta sekä ympäristöjen hallintaa.

Konttitekнологia on täydellinen kumppani DevOpsin tavoitteisiin, sillä se nopeuttaa toimitusta, helpottaa yhteistyötä tiimien välillä, vähentää manuaalista työtä ja lisää sovellusten luotettavuutta.

5.10 Kurssi 10: Application Development using Microservices and Serverless

Osa 1.

Aloitan uuden kurssin, joka käsittelee mikropalveluita ja palvelimettomuutta. Ensimmäisessä osassa tutustutaan mikropalveluiden taustaan, käyttötarkoitukseen ja metodeihin. Osa koostuu videoluennosta, luettavasta teoriasta, labrasta, yhteenvedosta ja monivalintakokeesta. Videoluennot ovat lyhyitä ja ytimekkäitä ja labrassa tutustutaan kuvitteellisen nettisivun modernisointiin. Osan suorittaminen oli sujuvaa enkä törmännyt ongelmiin. Monivalintakoe meni läpi ensimmäisellä yrittämällä, vaikka kysymykset olivat mielestäni vaikeita.

Osa 2.

Toisessa osassa tutustutaan REST arkkitehtuuriin ja tehdään oma REST rajapinta Flaskilla. Osa koostuu videoluennosta, labroista ja monivalintakokeesta. Labrat tehdään Courseran alustan tarjoamassa virtuaaliympäristössä ja niissä harjoitellaan Rest rajapinnan koodaamista, CRUD kutsuja ja testaamista Swaggerin avulla. Teoriaosuus pohjusti labroja kattavasti ja labrojen ohjeistus toimi hyvin. Osan suorittaminen oli sujuvaa ja kertasin teoriaosuudet ennen monivalintakoetta, joka meni läpi ensimmäisellä yrittämällä.

Osa 3.

Kolmas osa esittelee palvelimettomia sovelluksia ja vertaa niitä kontteihin ja perinteisiin malleihin. Osa koostuu videoluennosta, luettavasta teoriasta ja monivalintakokeesta. Osa ei sisältänyt labroja, ja suoritin sen nopeasti käymällä teoriaosuudet kahteen kertaan läpi.

Osa 4.

Neljännessä osassa tutustutaan IBM Cloud Code Engineen ja luodaan konttisovellus Dockerfilestä. Osa koostuu videoluennosta, kahdesta labrasta, yhteenvedosta ja monivalintakokeesta. Ensimmäisessä labrassa luodaan konttisovellus ja toisessa labrassa harjoitellaan sen

päivittämistä, skaalaamista ja käyttöönottoa. Osa sisälsi paljon uutta teoriaa ja labrat olivat hyvin mielenkiintoiset. Kertasin teoriaosuudet ja monivalintakoe meni läpi täysin pistein.

Osa 5.

Viimeinen osa koostuu vertaisarvioitavasta projektista, missä tehdään palvelimeton nettisovellus. Kurssin aikaisemmat osat ovat antaneet hyvän pohjan projektin tekoon ja ohjeistus on laadukas ja sitä on helppo seurata. Projekti tehtiin Courseran alustan virtuaaliympäristössä ja sen tekeminen onnistui sujuvasti ilman ongelmia. Projekti palautettiin vertaisarvioitavaksi Coursera alustalle henkilökohtaisen repositorion linkkinä. Lisäksi osan suorittaminen edellytti kahden vertaisarvioinnin tekemistä kanssaopiskelijoiden projekteille.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana ja käytin siihen noin 20 tuntia. Kurssi oli toimiva kokonaisuus ja nautin sen tekemisestä. Neljä ensimmäistä osaa antoivat hyvän teoreettisen pohjan vertaisarvioitavalle projektille. Vertaisarvioitava projekti on erinomainen keino saada käytännön kokemusta opitun teorian lisäksi ja henkilökohtaisen repositorion käyttö antaa konkreettisen todistuksen tehdystä työstä. Koen että kurssi syvensi ymmärrystäni kokonaisvaltaisesta sovelluskehittämisestä. Ymmärrys siitä, mikä tekniikka on paras missäkin yhteydessä on äärimmäisen tärkeä taito työelämässä.

Mikropalvelu on arkkitehtuurimalli, jossa sovellus jaetaan pieniin ja itsenäisiin palveluihin, jotka kommunikoivat keskenään rajapintojen avulla. Sovellusta, joka koostuu mikropalveluista, on helpompaa ja joustavampaa kehittää, ylläpitää ja skaalata. Kehitystyö ei kohdistu koko sovellukseen vaan erilliseen pieneen osaan siitä. Tämä kulkee käsi kädessä DevOps kehitysmallin tavoitteiden kanssa nopeuttaen kehitystä, lisäten joustoa ja parantaen sovelluksen luotettavuutta.

5.11 Kurssi 11: Introduction to Test Driven Development

Osa 1.

Aloitan uuden kurssin, joka käsittelee TDD kehitysmenetelmää ja testausta. Ensimmäinen osa käsittelee testauksen tärkeyttä sovelluskehityksessä. Osa koostuu videoluennosta, yhteenvedosta

ja monivalintakokeesta. Osa suorittaminen kävi nopeasti ja sujuvasti. Monivalintakoe oli helppo, aihe ei ole monimutkainen.

Osa 2.

Toisessa osassa esitellään TDD kehitysmenetelmää ja sen metodeja. Osa koostuu videoluennosta, kolmesta labrasta, yhteenvedosta ja monivalintakokeesta. Labroissa harjoitellaan yksikkötestausta pythonilla sekä testien koodaamista. Labrat tehdään Courseran alustan tarjoamassa virtuaaliympäristössä. Osa oli mielenkiintoinen ja labrat olivat hyvin ohjeistettuja.

Osa 3.

Kolmannessa osassa keskityttiin testien kattavuuteen ja mockaukseen. Osa koostuu videoluennosta, neljästä labrasta ja monivalintakokeesta. Labroissa harjoitellaan testikattavuutta, datan käyttöä testauksessa, datan mockaamista ja kaiken kurssilla opitun yhdistämistä TDD kehitysmenetelmän avulla. Labrat olivat hyvin ohjeistettuja ja niitä oli helppo seurata. Osan suorittaminen vei melko pitkään, koska se sisälsi niin monta labraa. Pidin osaa silti mielenkiintoisena juuri sen takia.

Osa 4.

Viimeinen osa sisältää vain koko kurssin kattavan monivalintakokeen. Kertaan kurssin teoriaosuudet rauhassa ja ajatuksella. Samalla mietin pitäväni enemmän kursseista missä aihe on rajattu, koen sen auttavan asian sisäistämässä. Läpäisen kokeen ensimmäisellä yrittämällä.

Kurssin yhteenveto.

Suoritin kurssin yhden päivän aikana ja se vei noin 10 tuntia. Kurssin aihe oli hyvin rajattu ja teoriaosuus hyvin kattava. Kurssin suorittaminen oli sujuvaa enkä kohdannut ongelmia sen aikana. Videoluennot olivat lyhyitä ja ytimekkäitä ja labrat hyvin ohjeistettuja. Coursera alustan tarjoama virtuaaliympäristö labroja varten toimii hyvin. Kehitystyö valmiissa ja tuetussa ympäristössä on helppoa ja vaivatonta. Kurssi syvensi ymmärrystäni testauksen tärkeydestä sovelluskehityksessä ja tutustuin ensimmäistä kertaa TDD kehitysmenetelmään, mitä pidin mielenkiintoisena. Haluan ehdottomasti kokeilla TDD kehitysmenetelmää kurssin ulkopuolella, ajatus siitä, että testit kirjoitetaan ennen koodia, tuntuu oudolta.

TDD on yleinen käytäntö DevOps kehitysmallin lisänä, koska se tukee jatkuvan testauksen ja jatkuvan integraation vaiheita. Automatisoitu testaus vaatii testien olemassaolon. Testit jäävät

monesti pienemmälle huomiolle sovelluskehityksessä kehittäjien keskittyessä sovelluksen koodiin. TDD kääntää asetelman toisinpäin, ensiksi kirjoitetaan testit toivotulle toiminnallisuudelle ja sen jälkeen toteuttava koodi, joka läpäisee ne. TDD:n avulla sovelluksen koodi muodostuu testien pohjalta jatkuvan palautteen kera. Tämä edistää laatua ja varmistaa että koodi toimii halutulla tavalla.

5.12 Kurssi 12: Continuous Integration and Continuous Delivery

Osa 1.

Aloitetaan uuden kurssin, joka käsittelee jatkuvaa integraatiota ja jatkuvaa julkaisua sekä CI/CD-putken rakentamista erilaisten työkalujen avulla. Ensimmäinen osa esittelee jatkuvan integraation ja jatkuvan julkaisun yhtenäisenä mallina. Osa koostuu videoluennosta, yhteenvedosta ja monivalintakokeesta. Kurssin aihe on selkeästi rajattu ja äärimmäisen tärkeä osa DevOps kehitysmallia. Kertaan teoriaosuudet rauhallisesti keskittyen ja läpäisen monivalintakokeen ensimmäisellä yrittämällä.

Osa 2.

Toinen osa esittelee jatkuvan integraation omana vaiheena ja käy läpi eri työkaluja millä sen voi toteuttaa. Osa koostuu videoluennosta, monivalintaharjoituksista, labrasta ja monivalintakokeesta. Labrassa harjoitellaan jatkuvan integraation automatisointia GitHub Actions työkalulla. Muutama teoriaosuus pohjaa aikaisempiin kursseihin. Kertaan teoriaosuudet ja läpäisen monivalintakokeen ensimmäisellä yrittämällä.

Osa 3.

Kolmas osa esittelee jatkuvan julkaisun omana vaiheena. Osa koostuu videoluennosta, monivalintaharjoituksista, viidestä labrasta, yhteenvedosta ja monivalintakokeesta. Labroissa harjoitellaan CI/CD-putken käyttöä, ja rakennetaan oma putki jatkuvalla julkaisulla käyttäen Tekton työkalua. Labrat ovat hyvin ohjeistettuja ja niiden suorittaminen on sujuvaa. Kertaan osion teoriaosuudet rauhallisesti ja keskittyneesti. Monivalintakoe oli haastava, läpäisin sen toisella yrittämällä.

Osa 4.

Viimeinen osa pitää sisällään koko kurssin kattavan monivalintakokeen. Kertaan kurssin teoriaosuudet, ja läpäisen monivalintakokeen. Monivalintakoe oli haastava yksityiskohtaisten kysymysten takia, sain pisteytyksenä 80 % mihin olen kuitenkin tyytyväinen.

Kurssin yhteenveto.

Suoritin kurssin yhden päivän aikana ja se vei noin 10 tuntia. Kurssi aihe oli rajattu hyvin ja sen ansiosta teoriaosuus oli hyvin kattava ja syväluotaava. Aiheen esitleminen kokonaisuutena, sekä omina vaiheinaan, antoi minulle vahvan ymmärryksen CI/CD-putken suunnittelusta, rakenteesta ja implementaatiosta. Labrat olivat hyvin ohjeistettuja ja niiden suorittaminen oli toteutettu hyvin. Varsinkin CD-putken tekeminen kolmannen osan labroissa oli opettavainen kokemus. Jatkuva integraatio ja jatkuva julkaisu on yhden tärkeimmistä DevOps kehitysmallin vaiheista. Vaikka CI/CD-putki esittää ne saman asian yhteydessä, on tärkeää ymmärtää, että ne ovat täysin omat vaiheensa. Kurssin edellytyksenä oli ymmärrys pilvipalveluista, DevOps periaatteista, konteista ja Kubernetesista, Linux komennoista ja versionhallinnasta Gitin ja GitHubin avulla.

5.13 Kurssi 13: Application Security and Monitoring

Osa 1.

Aloitan uuden kurssin, joka käsittelee tietoturvaa ja monitorointia sovelluskehityksessä. Kurssi pohjaa aikaisemmin suoritettuihin kursseihin keskittyen tietoturvan rooliin. Ensimmäinen osa esittelee tietoturvan osuutta sovelluskehityksen elinkaareissa ja DevSecOps kehitysmallia. Osa koostuu videoluennosta, monivalintaharjoituksista ja monivalintakokeesta. Videoluennot ovat lyhyitä ja esittävät aiheen kattavasti. Hyvin rajatun aiheen vuoksi uutta tietoa ei ole liikaa ja se on helppo sisäistää. Kertaan osan teoriaosuudet ja läpäisen monivalintakokeen ensimmäisellä yrittämällä.

Osa 2.

Toinen osa käsittelee testauksen roolia sovelluskehityksen tietoturvassa. Osa koostuu videoluennosta, kolmesta labrasta, yhteenvedoista, monivalintaharjoituksista ja monivalintakokeesta. Labroissa työskennellään Courseran alustan tarjoamassa virtuaaliympäristössä ja harjoitellaan staattista sekä dynaamista analyysia python projektin parissa. Lisäksi harjoitellaan haavoittuvuuksien havaitsemista ja kirjastojen skannaamista

haavoittuvuuksien varalta. Videoluennot esittävät teorian kattavasti, labrat ovat hyvin ohjeistettuja ja osan suorittaminen on sujuvaa.

Osa 3.

Kolmannessa osassa tutustutaan OWASP Top 10 listaan, missä esitellään kymmenen yleisintä haavoittuvuutta sovelluksissa. Osa koostuu videoluennoista, kolmesta labrasta ja monivalintakokeesta. Labroissa harjoitellaan haavoittuvuuksilta suojautumista. Osan suorittaminen oli sujuvaa ja mielenkiintoista. Osa sisälsi paljon uutta tietoa ja kertasin teoriaosuudet tarkasti uudelleen ennen monivalintakoetta. Monivalintakoe oli haastava ja kysymykset hyvin tarkkoja, läpäisin sen kolmannella yrittämällä.

Osa 4.

Neljäs osa esittelee parhaita käytäntöjä tietoturvan suhteen. Osa koostuu videoluennoista, kahdesta labrasta ja monivalintakokeesta. Labroissa harjoitellaan turvallisen koodin kirjoittamista ja kehitysympäristön turvaamista. Videoluennot esittivät teorian kattavasti ja labrat olivat hyvin ohjeistettuja, mikä teki osan suorittamisesta sujuvaa. Osa oli lyhyt ja tiivis paketti rajatun aiheen vuoksi.

Osa 5.

Viidennessä osassa tutustutaan monitorointiin tietoturvan apuna. Osa koostuu videoluennoista, monivalintaharjoituksista ja monivalintakokeesta. Osa on lyhyt rajatun aiheen vuoksi ja helppo sisäistää. Kertaan teoriaosuudet ennen monivalintakoetta ja läpäisen sen ensimmäisellä yrittämällä.

Osa 6.

Kuudennessa osassa tutustutaan monitoroinnissa käytettyihin työkaluihin ja tekniikoihin. Osa koostuu videoluennoista, labroista, monivalintaharjoituksista, yhteenvedosta ja monivalintakokeesta. Labroissa harjoitellaan monitorointia käyttäen työkaluina Prometheusta ja Grafanaa. Osan suorittaminen kävi sujuvasti ja labrat olivat hyvin ohjeistettuja.

Osa 7.

Viimeinen osa sisältää lyhyen esittelyn loggauksesta ja koko kurssin kattavan monivalintakokeen. Kurssi sisälsi paljon uutta tietoa ja kertaan kurssin teoriaosuudet huolella. Monivalintakoe on

haastava, kysymyksiä on 20 ja ne ovat erittäin yksityiskohtaisia. Läpäisen monivalintakokeen toisella yrittämällä ja saan täydet pisteet, johon olen erittäin tyytyväinen.

Kurssin yhteenveto.

Suoritin kurssin yhden viikon aikana ja käytin siihen noin 20 tuntia. Kurssin aihe oli rajattu tietoturvan osaan sovelluskehityksessä ja kurssin pohjana toimi aikaisemmin suoritettut kurssit. Aiheen rajaus mahdollisti syvällisen perehtymisen tietoturvaan, sen tärkeyteen ja työkaluihin millä sitä implementoidaan sovelluksen elinkaaren eri vaiheisiin. Kurssin toteutus oli laadukas, teoria esitettiin kattavasti ja labrat olivat hyvin ohjeistettuja. Kurssin suorittaminen oli sujuvaa enkä kohdannut ongelmia. Kurssi antoi minulle ymmärryksen tietoturvasta tärkeänä osana sovelluskehityksen elinkaarta. Suoritin harjoitteluni finanssi- ja vakuutusalan yrityksessä, missä tietoturva on erityisen tärkeässä roolissa. Harjoitteluni takia pidin kurssin opetuksia itsestäänselvyyksinä. Tietoturvan pitäisi kuulua olennaisena osana sovelluskehittäjän työnkuvaan ja tämä kurssi antaa siihen hyvän pohjan.

DevSecOps lisää tietoturvan DevOpsin kehitysmallin elinkaareen jatkuvan integraation ja automaation avulla. Tietoturva otetaan huomioon suunnittelusta ja kehittämisestä aina tuotantoon ja ylläpitoon asti. DevSecOps kehitysmallissa tietoturvaan liittyviä tehtäviä integroidaan esimerkiksi CI/CD-putkeen.

5.14 Kurssi 14: DevOps Capstone Project

Sertifikaattikonaisuuden viimeinen kurssi koostuu vertaisarvioitavasta loppuprojektista. Loppuprojekti kokoaa yhteen kaiken opitun aikaisemmilta kursseilta ja se tehdään kuudessa osassa.

Osa 1.

Ensimmäisessä osassa tehdään projektinhallintasuunnitelma. Osa koostuu labrasta ja yhteenvedosta. Labrassa tehdään projektinhallintasuunnitelma koko kurssille. Projektille luodaan repositorio henkilökohtaisen GitHub tilin alle ja se yhdistetään Zenhubin kanban tauluun. Labra on hyvin ohjeistettu ja sen suorittaminen on sujuvaa.

Osa 2.

Toisessa osassa tehdään REST rajapinta ja viedään projektia eteenpäin kanban taululla. Osa koostuu labrasta ja yhteenvedosta. Labrassa työskennellään Courseran virtuaaliympäristössä ja kehitetään REST rajapinta TDD mallin mukaan. Labran ohjeistus on selkeää enkä kohtaa ongelmia.

Osa 3.

Kolmannessa osassa repositorioon tehdään CI-putki ja viedään projektia eteenpäin kanban taululla. Osa koostuu labrasta ja yhteenvedosta. Labra on ohjeistettu hyvin mikä tekee työskentelystä sujuvaa.

Osa 4.

Neljännessä osassa sovellus pakataan konttiin Dockerilla ja käyttöön otetaan Kubernetesin avulla sekä päivitetään projektin kulkua kanban taululla. Osa koostuu labrasta ja yhteenvedosta. Labrassa työskennellään Courseran virtuaaliympäristössä ja se on ohjeistettu hyvin. Kohtaan ensimmäistä kertaa ongelmia virtuaaliympäristön kanssa, pidin ruokatauon ja ympäristö käynnistyi uudelleen. En joudu aloittamaan alusta, mutta joudun tekemään pari tehtävää uusiksi.

Osa 5.

Viidennessä osassa projekti viimeistellään CD-putken rakentamisella ja kanban taulun päivittämisellä. Osa koostuu labrasta ja yhteenvedosta. Labra on hyvin ohjeistettu ja siinä käytetään Tektonia CD-putken tekemiseen. Työskentely oli sujuvaa, ja muistin työntää koodin etärepositorioon talteen useaan kertaan ongelmien varalta. Projektin teko onnistui kokonaisuutena hyvin ja olen erittäin tyytyväinen lopputulokseen.

Osa 6.

Viimeinen osa koostuu projektin dokumentoinnista ja palauttamisesta vertaisarvioon. Dokumentointi tehtiin kuvankaappauksin, jotka palautettiin Coursera alustalle projektin repositorion linkin kanssa. Lisäksi vertaisarvioin kahden kanssaopiskelijan loppuprojektit. Vertaisarviointi oli mielenkiintoista, sain nähdä vähän erilaisia ratkaisuja projektin toteuttamiseen.

Kurssin yhteenveto.

Suoritin loppuprojektin kahden päivän aikana ja käytin siihen noin 20 tuntia. Projektin teko oli jaettu viiteen labraan, jotka koostuivat projektin eri osien tekemisestä. Tärkein asia projektissa ei ollut itse

sovellus, vaan DevOps kehitysmallin mukainen toteutus. Loppuprojekti kokosi sertifikaatin sisältämät kurssit yhdeksi kokonaisuudeksi. Labrassa kokemani ongelma johtui siitä, että en käynnistänyt virtuaaliympäristöä uusiksi aloittaessani neljännen osan tekemistä. Osat on suunniteltu tehtäviksi erillään, mutta yritin tehdä projektin yhden päivän aikana. Luovutin 16 tunnin jälkeen ja viimeistelin projektin seuraavana päivänä. Henkilökohtaisen repositorion käyttö antaa minulle konkreettiset todisteet tehdystä DevOps kehitysmallin mukaisesta sovelluskehitystyöstä. Loppuprojekti oli erinomainen päätös sertifikaatin suorittamiseen. Esitän pohdinnassa lyhyen yhteenvedon kaikista kursseista kokonaisuutena.

6 POHDINTA

Opinnäytetyön tavoitteena oli pohtia Courseran sertifikaatin suorittamista Tietojenkäsittelyn tutkinto-ohjelman osana ja suorittaa IBM DevOps and Software Engineering Specialization sertifikaatti sekä arvioida sen hyötyä minulle opiskelijana. Kurssien yhteenvedoissa kuvasin kurssin aihetta, tehtäviä, suorittamisen sujuvuutta, kohtaamiani ongelmia sekä reflektoin kurssin sisältöä teemaan ja tietoperustaan. Lisäksi pohdin omaa kehitystäni aloittelevana sovelluskehittäjänä ja kurssien minulle tuomaa hyötyä.

Sertifikaatin suorittaminen oli erittäin antoisa, opettavainen ja mielenkiintoinen kokemus, jonka tekemiseen minulta meni noin kaksi kuukautta. Kurssit muodostivat laajan kokonaisuuden sovelluskehityksestä DevOps kehitysmallin mukaisesti. Suurin osa kurssien teoriasta oli videoluentojen muodossa, ja ne olivat lyhyehköjä mutta kattavia tiivistelmiä aiheesta. Oppimista tuettiin monivalintatehtävin ja yhteenvedoin, jotka kertasivat videoluentojen teoriaa. Labroissa työskenneltiin selaimesta avattavassa virtuaaliympäristössä, joka oli valmis ja kattavasti ohjeistettu ympäristö kyseistä labraa kohden. Tämä teki harjoitusten ja projektien tekemisestä sujuvaa ja mielekästä. Koko kurssien kattavat monivalintakokeet eivät olleet helppoja, vaan ne vaativat kurssin teoriaan perehtymistä. Kysymykset olivat tarkkoja ja kattoivat aiheen hyvin ja jos monivalintakoe piti uusia, kysymykset vaihtuivat. Muutamassa kurssissa arvostelu tehtiin vertaisarvioitavan projektin perusteella. Vertaisarvioitavat projektit tehtiin labrojen tapaan virtuaaliympäristöissä ja ne tallennettiin omaan henkilökohtaiseen GitHub repositorioon. Vertaisarviointi antoi näkökulmaa erilaisiin ratkaisuihin ja projektien tallentaminen omaan repositorioon on hyvä tapa näyttää osaamista ja kokemusta sovelluskehityksessä.

Päiväkirjan kirjoittaminen toimi hyvänä keinona sisäistää erilliset kurssit osana DevOpsia. Tietoperustan kirjoittaminen, lähteiden haku ja kurssien peilaaminen teemaan syvensi ymmärrystäni kokonaisvaltaisesta sovelluskehityksestä sekä antoi minulle vahvan pohjan DevOps kehitysmallista, sen eri osista ja implementoinnista. Tietoperustan kirjoittaminen oli haastavaa lähinnä teeman laajuuden takia. Onnistuin omasta mielestäni kuvaamaan DevOpsin laajasti ja helposti ymmärrettävästi.

Sertifikaatin suorittaminen osana opintoja oli todella hyvä lisä opintosuunnitelmaani. Se antoi minulle mahdollisuuden erikoistua aiheeseen, joka on erittäin haluttu taito työmarkkinoilla.

Sertifikaatti ja repositioon lisätyt projektit toimivat todistuksena hankitusta taidosta ja kokemuksesta. Näillä voi olla iso merkitys harjoittelupaikan saamisessa tai työelämään siirtymisessä. Toinen hyöty tulee sertifikaattien työelämälähtöisyydestä. Sovelluskehitysprosessin ymmärtäminen kokonaisuutena on hyvä keino pienentää opintojen ja työelämän välistä kuilua. Suosittelen sertifikaatin suorittamista lämpimästi jokaiselle Tietojenkäsittelyn opiskelijalle, jota kiinnostaa kokonaisvaltainen sovelluskehitys.

Koen että tästä opinnäytetyöstä voi olla hyötyä sovelluskehityksestä kiinnostuneille opiskelijoille. Olen nyt suorittanut kaksi Coursera sertifikaattia osana Tietojenkäsittelyn tutkinto-ohjelmaa. Selkeä etu mitä ne ovat minulle tarjonneet on erikoistuminen sellaisiin aiheisiin, mihin en olisi muuten pystynyt erikoistumaan. Lisäksi voin näyttää opitut taidot ja kokemuksen tehdyillä projekteilla repositoriossani. Sertifikaatteja kannattaisi suositella jo ensimmäisen vuoden opiskelijoille, jos aihe heitä kiinnostaa. Se auttaisi heitä hahmottamaan sovelluskehitystyötä kokonaisuutena ja saattaisi selkeyttää heidän tavoitteitansa opintojen suhteen.

LÄHTEET

Atlassian 2023. What Is DevOps? Hakupäivä 10.7.2023. <https://www.atlassian.com/devops>.

Boersma, Eric 2022. Is There a Place for the Waterfall Methodology in 2021? Hakupäivä 24.7.2023. <https://www.plutora.com/blog/waterfall-methodology>.

Bose, Shreya 2023. What is DevOps? (A Beginner's Guide). Hakupäivä 10.7.2023. <https://www.browserstack.com/guide/devops-for-beginners>.

Das, Sourojit 2023. DevOps Lifecycle: Different Phases in DevOps. <https://www.browserstack.com/guide/devops-lifecycle>.

Gitlab 2023. What is DevOps? Hakupäivä 1.7.2023. <https://about.gitlab.com/topics/devops/>.

Hamilton, Thomas 2023. Agile vs Waterfall – Difference Between Methodologies. Hakupäivä 10.7.2023. <https://www.guru99.com/waterfall-vs-agile.html>.

IBM 2023. What is DevOps? Hakupäivä 10.7.2023. <https://www.ibm.com/topics/devops>.

IPM 2022. Waterfall Methodology. Hakupäivä 1.7.2023. <https://www.projectmanagement.ie/blog/waterfall-methodology/>.

Jain, Aditi 2023. DevOps Lifecycle: Definition, Phases. Hakupäivä 8.7.2023. <https://www.knowledgehut.com/blog/devops/devops-lifecycle>.

KnowledgeHut 2023. 8 Key Challenges Of Implementing DevOps And Overcoming Them. Hakupäivä 26.6.2023. <https://www.knowledgehut.com/blog/devops/devops-implementation-challenges>.

Mohanan, Remya 2022. What Is DevOps Lifecycle? Hakupäivä 8.7.2023. <https://www.spiceworks.com/tech/devops/articles/what-is-devops-lifecycle/>.

Moustakis, Ioannis 2023. 16 DevOps Best Practices Every Developer Should Know. Hakupäivä 1.7.2023. <https://spacelift.io/blog/devops-best-practices>.

Ohjelmistokehityksen menetelmät 2021. Opas Gitin perusteisiin. Hakupäivä 10.7.2023. <https://book.sovelluskontti.com/versionhallinta/opas-gitin-perusteisiin>.

Thinking Portfolio 2016. Projektien vesiputousmalli ja sen viisi heikkoutta. Hakupäivä 1.7.2023. <https://thinkingportfolio.com/projektien-vesiputousmalli-ja-sen-viisi-heikkoutta/>.

Valagroup 2020. Mitä DevOps on – Neljä näkökulmaa. Hakupäivä 1.7.2023. <https://www.valagroup.com/fi/blogi/mita-devops-on-nelja-nakokulmaa/>.

Vaniukov, Slava 2021. Best of 2021 – How to Combine DevOps and Agile. Hakupäivä 26.6.2023. <https://devops.com/how-to-combine-devops-and-agile/>.