



Kehittäminen pilviympäristössä:

D365FO ohjelmistokehittäjän päiväkirja

Juha-Matti Aho

Haaga-Helia ammattikorkeakoulu

Tietojenkäsittelyn tradenomi

Amk-opinnäytetyö

2022

Tiivistelmä

Tekijä(t) Juha-Matti Aho
Tutkinto Tradenomi
Opinnäytetyön nimi Kehittäminen pilviympäristössä: D365FO ohjelmistokehittäjän päiväkirja
Sivu- ja liitesivumäärä 51 + 3
<p>Tässä päiväkirjamuotoisessa opinnäytetyössä kuvataan uuden työntekijän tutustumista D365FO ohjelmistokehittäjän tehtäviin. Tekijä työskentelee kokopäiväisesti suuressa ICT-alan yrityksessä. Seuranta toteutettiin ajalta 04.04.2022-29.05.2022. Päiväkirjan kirjoittaminen alkaa ajalta, kun työntekijää vielä koulutetaan tehtävään ja seurantajakson aikana hän pääsee virallisiin asiakasprojekteihin. Opinnäytetyön tarkoituksena oli kuvata uusien työtehtävien, ohjelmistojen ja pilviympäristön oppimista ja niissä kehittymistä. Oppiminen kuvataan päivittäisinä raportteina ja viikoittaisissa analyyseissa siihen syvennyttään teorian kautta.</p> <p>Opinnäytetyön johdannossa esiteltiin työn tavoitetta, ammattikäsitteitä ja keskeistä tietoperustaa. Lähtötilanteen kuvaksessa tekijä analysoi hänen nykyistä työtilannettansa, työpaikan sidoryhmiä sekä työpaikan vuorovaikutustilanteita. Nykytilan analyysissa kerrottiin keskeisistä D365FO kehittäjän työtehtävistä, sekä käytetyistä teknologioista.</p> <p>Opinnäytetyön päiväkirjaraportoinnissa kuvattiin D365FO-ohjelmoijan varsinaista työtehtävää, käytettäviä sovelluksia ja päivittäisiä kokemuksia. Päivittäisissä raporteissa esiteltiin työpäivien aikana suoritettuja tehtäviä ja käytettyjä tekniikoita. Viikoittaisissa analyyseissa keskityttiin pääaiheiden mukaiseen tehtävien analysoimiseen lähteitä apuna käyttäen.</p> <p>Pohdinnat ja päätelmät -osiossa verrattiin nykyistä tilannetta opinnäytetyön alussa kuvattuun lähtötilanteeseen. Osiossa käsiteltiin, kuinka tekijä on kehittynyt määrittelemässään pääaiheissa, mikä tilanne on nykyään ja mitä kehitettävää tulevaisuudessa voisi vielä olla. Tekijä myös pohtii sitä, mitä päiväkirjamuotoinen opinnäytetyö on opettanut ja kuinka se on tukenut työtehtävissä kehittymistä.</p>
Asiasanat Ohjelmistokehitys, ICT, toiminnanohjausjärjestelmät, Microsoft

Sisällys

1	Johdanto.....	1
1.1	Ammattikäsitteet.....	2
2	Lähtötilanteen kuvaus	4
2.1	Oman nykyisen työ analysointi.....	4
2.2	Sidosryhmien esittely	5
2.3	Työpaikan vuorovaikutustilanteet.....	7
3	Seurantajakson raportointi viikkoanalyysineen	9
3.1	Seurantaviikko 1	9
3.2	Seurantaviikko 2	13
3.3	Seurantaviikko 3	19
3.4	Seurantaviikko 4	22
3.5	Seurantaviikko 5	27
3.6	Seurantaviikko 6	32
3.7	Seurantaviikko 7	36
3.8	Seurantaviikko 8	43
4	Pohdinta ja päätelmät.....	49
	Lähteet.....	52

1 Johdanto

D365FO kehittäjä on varsin kuvaava termi työlleni, mutta monelle, kuten itsellenikään se ei aluksi kertonut yhtään mitään. Opinnäytetyöni pyrkii kuvaamaan omaa oppimistani ja kehittymistäni tässä työtehtävässä, sekä avaamaan samanaikaisesti työtehtävääni ja toimintaympäristöä. Seuranta-jakso opinnäytetyölle on kahdeksan viikkoa ajalta 04.04.2022-29.05.2022. Raportointia toteutetaan päivittäin työtehtävien ja tapahtumien kuvaamisella sekä viikoittaisilla analyyseilla. Viikoittaisissa analyyseissa poimitaan viikon tärkeimpiä asioita ja niitä analysoidaan teoriapohjaa hyväksikäyt-täen. Tällä tavalla tekijä pääsee syventymään tärkeisiin aiheisiin ja oppimaan työtehtävälle tärkeitä asioita. Yrityksen ainoana vaatimuksena oli se, että asiakkaiden nimet tai heidän henkilökohtaiset tietonsa eivät tule julki.

Työskentelen suuressa IT-alan yrityksessä nimeltään Fellowmind Finland Oy Ab. Fellowmind on suuri konserni, jolla on toimipisteitä kuudessa Euroopan maassa. Suomessa toimistoja on seitse-män ja itse työskentelen Espoon toimipisteellä. Yritys on jaettu toiminnaltaan kuuteen eri osastoon, josta omani on Cloud Solutions. Fellowmind tarjoaa moderneja liiketoimintajärjestelmiä, pilvi- ja tie-toturvaratkaisuja, datan ja tekoälyn sekä modernin työskentelyn asiantuntijuutta. Fellowmind on Microsoftin yksi Euroopan johtavista Microsoft-kumppaneista. Oma työskentelyni on Microsoftin D365 Finance and Operations toiminnanohjausjärjestelmien kehittäjänä.

Aloitin työt yrityksessä 14.03.2022. Ensimmäiset viikot olivat yrityksen toimintoihin tutustumista, jonka jälkeen minulle alettiin kouluttaa omaa työtehtävääni. Aloitin päiväkirjan pitämisen silloin, kun koulutus oli jo alkanut, joten minulla oli enemmän kerrottavaa varsinaisesti kehittymisestä. D365FO kehittäminen on aihe, jota ei missään opeteta, joten jokainen, joka aloittaa tämän tehtävän, on käy-tävä läpi koulutus. Työtehtävässäni tulen tarvitsemaan omaehtoista asennetta ja pitkäjänteisyyttä oppiakseni vaadittavat asiat. Tämän lisäksi vahvat sosiaaliset taidot ovat eduksi, sillä minun täytyy toimia tiimissä ja kehitykseni tukena on oma mentori, jonka kanssa kommunikaation on oltava sel-vää.

Asetin tavoitteekseni oppia pilviympäristön, FO-ympäristön ja kehittyä tiimityöskentelijänä ohjel-mointitiimissä. Pilviympäristö, jossa työskentelemme, koostuu kokonaan Microsoft Azuren palve-luista. Tutustun laajasti eri työkaluihin, mutta tärkeimpinä itselleni ovat Azure DevOps ja virtuaaliko-neet. FO-ympäristö käsittää sisäänsä valtavan tietokannan, Visual Studion, uuden koodikielen X++ ja itse sovelluksen käyttöliittymän. Työskentely IT-alalla on itselleni kokonaan uutta ja haluan kes-kittyä erityisesti tiimityöskentelyyn ja etätyöajan tuomiin muutoksiin.

Opinnäytetyön tietoperusta on koostettu eri nettilähteistä ja kirjoista. Työtehtäväni perustuu Microsoftin työkaluihin ja ne päivittyvät niin usein, että varmin tapa saada uusin tieto, on hakea se heidän omilta sivuiltaan. Käyttämäni ohjelmisto on Microsoftin Finance and Operations ja käytän sen omaa virallista dokumentointia nimeltään Finance and Operations application documentation (Microsoft 2022a). Tämän lisäksi käytän useita muita Microsoftin lähteitä, mutta nostaisin toiseksi tärkeimmäksi lähteeksi Microsoft Learnin (Microsoft 2022b). Learn sisältää opintokursseja useille Microsoftin työkaluille ja sovelluksille, ja itse käyn osana työtäni yhtä näistä kursseista läpi. Tämä kurssi opastaa FO:n käyttöön ja johtaa lopulta sertifiointikokeeseen. Tämän lisäksi käytän muita nettilähteitä koodaamisen tueksi ja sovellusten käytön teoriaan.

Tiimi- ja projektityöskentelystä valitsin lähteekseni parhaaksi kokemani teoksen, joka selkeitten kuvaa näitä aiheita. Tämä teos on Mikko Mäntynevan Hallittu projekti – Jäntevästä suunnittelusta menestykselliseen toteutukseen (2016). Mäntynevan kirjassa kuvataan tiimityöskentelyn ja projektin eri vaiheita tarkasti, antaen hyvän näkemyksen onnistuneista projekteista ja yleisimmistä sudenkuopista. Tämän avulla pystyn analysoimaan työtiimini toimintaa ja liittymään siihen mukaan. Tämän lisäksi käytän muita lähteitä ohjelmoijan työn tekemiseen ja mentorin kanssa toimimiseen.

1.1 Ammattikäsitteet

Lista työn keskeisimmistä ammattikäsitteistä, joita tulen käyttämään opinnäytetyössä. Joitakin yksittäisiä käsitteitä tulen avaamaan itse päiväkirjan tekstissä.

API eli ohjelmointirajapinta. Rajapinnan kautta saadaan tai lähetetään tietoa.

Azure on Microsoftin pilvipalvelualusta.

AOT eli Application object tree on D365FO-ympäristön oma tietokantarakenne.

Confluence on Atlassianin tarjoama työpaikan sisäinen nettisivu sisäiselle informaatiolle ja dokumentoinnille.

D365FO tai **FO** eli Microsoft Dynamics 365 Finance and Operations on toiminnanohjausjärjestelmä.

DevOps on Azuren tarjoama palvelu, joka yhdistää koodin versiohallinnan, rakentamisen ja julkaisun samalla alustalla.

ERP eli Enterprise Resource Planning, suomeksi toiminnanohjausjärjestelmä, on yrityksen eri toimintoja yhdistävä tietojärjestelmä. ERP:in avulla voidaan ohjata toimintoja kuten varastonhallintaa, tuotantoa ja laskutusta samassa järjestelmässä.

Jira on Atlassianin omistama tehtävienhallintaohjelmisto. Jira:ssa perustetaan ja seurataan tikettejä, jotka vastaavat yksittäisiä töitä tai tehtäviä.

ICT eli Information and communications technology on termi, jolla kutsutaan yleisesti koko tietojenkäsittelyn alaa.

Postman on sovellus, jolla voidaan suorittaa kutsuja eri rajapintoihin.

Slack on keskustelusovellus, joka toimii työpaikan sisäisessä viestinnässä.

Solaforce on HR-järjestelmä, josta näemme henkilökohtaiset työntekijän tiedot.

Tiketti on yleistermi tehtäville IT-järjestelmissä ja ohjelmistoissa.

Teams eli Microsoft Teams on sovellus videotapaamisten ja keskustelujen pitämiseksi.

Visual Studio on Microsoftin kehitystyökalu, jossa voit kirjoittaa koodia.

Virtuaalitietokone on pilvipalvelimelle perustettu tietokone, johon voit ottaa etänä yhteyttä.

X++ on D365FO-järjestelmän oma käyttämä koodikieli.

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työ analysointi

Aloitin työssäni 14.03.2022 eli ennen opinnäytetyön aloittamista olen ehtinyt olla varsinaisessa työssä kolme viikkoa. Toimenkuvani ei ollut minulle entuudestaan tuttu, mutta kolmen viikon aikana olen saanut hyvän organisaatiosta ja omasta roolistani. Toimin Microsoft Dynamics 365 Finance and Operations kehittäjänä. Toimenkuvaani kuuluu ERP-ratkaisujen kehittäminen ja ylläpito Azuren pilvipalveluympäristössä. Tämän lisäksi toimin tulevaisuudessa konsulttina asiakkaille ja olen osana noin kymmenen hengen tiimiä.

Työn alkuvaiheissa minua koulutetaan tehtävään noin kahdeksan viikon ajan, jonka jälkeen, tai jo koulutuksen aikana, minut laitetaan asiakasprojektiin. Koulutuksen alkuun minulle esiteltiin yrityksen toiminta, osastot ja toimintatavat, sekä perehdytettiin yleisiin käytäntöihin. Yleisiä työkaluja ovat muun muassa Jira, Power BI, LastPass, Solaforce, OData ja Dynamics 365 Business Central. Tämän jälkeen koulutettiin varsinaisiin sovelluksiin, joiden kanssa teen työni. Näitä ovat Azure, Azure DevOps, Visual Studio ja Microsoft Dynamics 365 Finance and Operations. Pääsin nopeasti näiden työkalujen pariin ja olen saanut harjoituksia, joilla olen tutustunut niiden toimintaan. Harjoituksiin liittyy ympäristön perustamista, muokkaamista, koodausta, versiohallintaa ja testausta.

Tarvitsen työssäni ratkaisukeskeisyyttä ja itsenäistä työskentelykykyä, mutta minulla on kuitenkin ympärilläni vahva tiimi ja mentori, jolta voin kysyä apua. Azuren ja virtuaalitetokoneiden kanssa työskentely ovat tuttuja, mutta niissä on melkein loputon määrä opiskeltavaa. Pääsääntöisenä koodikielenä työssäni on X++, josta minulla ei ole aiempaa kokemusta. Olen kuitenkin koodannut monella kielellä ja olio-ohjelmointi on minulle hyvin tuttua, joten X++n oppiminen ei tulisi olla liian haastavaa. Työpaikan opetus pohjautuu vahvasti Microsoftin Learn -portaalin oppitunteihin, joita voi käydä omalla ajallaan läpi ja lopulta on tarkoitus suorittaa Microsoft MB-500 sertifikaatti osaamisen tunnistamiseksi.

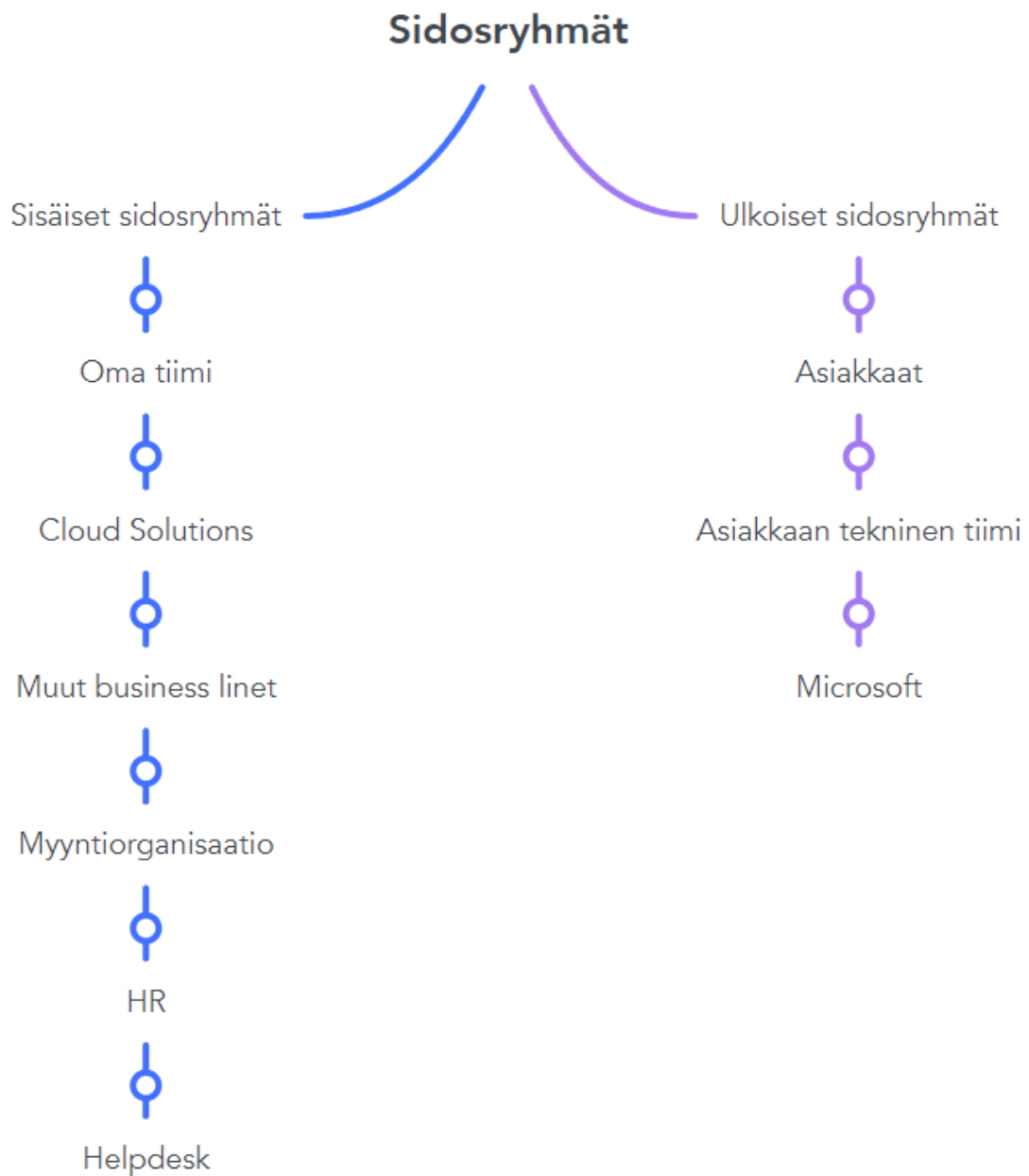
Aloitin uudessa työssäni tiedostaen, että minulta puuttuu vaadittavia taitoja, mutta sen vuoksi oppimiseen on varattu alusta kahdeksan viikkoa. Koen tämän päiväkirjan kirjoittamisen auttavan ja syventävän oppimistani samanaikaisesti. Työnantaja, kuin minäkin, olemme arvioineet, että osaamiseni on kuitenkin tarpeeksi vahvaa, että tulen suoriutumaan varsinaisessa tehtävässä hyvin. Koulussa suuntautumisvaihtoehtona olen erikoistunut ohjelmistotuotantoon, mutta olen käynyt melko laajasti infrastruktuurin kursseja tukeakseni osaamistani. Infrastruktuurin kursseista on selkeästi hyötyä uusien ympäristöjen käyttämisessä ja tämän lisäksi tutkintoon liittyvät liiketalouden opinnot ovat edukseen. Kehitysympäristö ja koodikieli ovat sellaisia, että niitä ei kouluissa opeteta ja työnantajakin tämä tiedostaa. Siksi jokainen koulutetaan tähän tehtävään erikseen.

Aiemmassa työelämässä olen ollut vuosia asiakaspalvelutyössä ja suorittavissa tehtävissä matkailualalla. Tämä ei suoranaisesti liity koodaamiseen, mutta konsultin tehtävissä ja yleisessä tiimityöskentelyssä kokemuksestani on hyötyä. Työelämässä oleminen ja vastuun kantaminen työntekijänä ovat minulle luonteista. Kommunikaatio- ja ihmistaitoni ovat hyvällä tasolla, jotka ovat mielestäni tärkeitä taitoja nykymailman ohjelmoijille. Pilvipalveluympäristössä työskentely on minulle luontaista ja koodaamistaitoni koulutuksen määrään nähden hyviä. Suurimman osan kursseista olen suorittanut kiitettävillä arvosanoilla. Työssä olen innoissani siitä, että saan vihdoin keskittyä yhteen tekniikkaan ja koodauskieleen, jolloin pystyn syventämään osaamistani seuraavalle tasolle.

Oma tasoni on aloitteleva toimija. Työssä suoriutuminen vaatii työkaverin antamaa ohjeistusta ja apua. Työpaikalla minulla on tämän takia mentori, jolta voin kysellä mitä tahansa työhön liittyvää. Itsenäinen ja joustava suoriutuminen ympäristössä on vajavaista, sillä minun on opeteltava useita eri sovelluksia ja asetuksia, mutta ohjelmointi on itselleni suhteellisen luontevaa. Suurin vaikeuteni työn alkuvaiheissa on ollut päivittäin vaihtuvat ohjelmistot, joita minulle on esitelty. Nyt kun pystyn keskittymään vain niihin, joita tulen käyttämään työssäni, uskon kehittyväni nopeasti itsenäisemmäksi suoriutujaksi.

2.2 Sidosryhmien esittely

Itseäni eniten koskettavat sidosryhmät ovat eritelty kuvassa 1. Tärkeitä sisäisiä sidosryhmiä ovat oma tiimi, cloud solutions, muut business linet, myyntiorganisaatio, hr ja helpdesk. Ulkoisissa sidosryhmissä ovat asiakkaat, asiakkaan tekninen osasto ja Microsoft.



Kuva 1. Sidosryhmien esittely

Oma tiimini on noin kymmenen hengen ryhmä ja olemme Cloud Solutions osaston alaisia. Muita yritykseni toimialoja on viisi. Heidän kanssaan saatan olla tekemisissä, mutta en niin säännöllisesti, jonka takia yhdistin ne yhdeksi. Myyntiorganisaatio on tärkeä projektien saamisessa ja toimintojen

myymisessä asiakasyrityksille ja välillä he saattavat tarvita apua teknisten asioiden konsultoinnissa, jossa itse olen osana. HR ja helpdesk ovat läsnä toimistolla ja auttavat henkilöstön ja teknii-
kan ongelmissa.

Ulkoisina sidosryhminä ovat asiakkaan ja asiakkaiden tekninen tiimi. Joudun konsultoimaan asiak-
kaita teknisissä ratkaisuissa ja keskustelemaan heidän teknisen tiiminsä kanssa siitä, mikä ratkaisu
olisi heille sopivin. Microsoft on mukana, sillä olemme Microsoftin kanssa läheinen yhteistyökump-
pani ja käytämme heidän työvälineitään. Joissakin teknisissä ratkaisuissa tarvitsemme heidän apu-
aan ja minun täytyy olla yhteyksissä heihin saadaksemme tiettyjä muutoksia läpi ohjelmistoissa.

2.3 Työpaikan vuorovaikutustilanteet

Työn aloituksessa käyn läpi ohjelmistoja kouluttajan kanssa. Hän näyttää minulle sovelluksia ja
käytäntöjä ja saan sen jälkeen tehtäviä tehtäväkseni. Koulutuksen aikana on tärkeä kysyä paljon ja
olla aktiivinen, sekä varsinaisessa työssä minun täytyy kysellä mentorilta ja muilta tiimiläisiltä pal-
jon kysymyksiä. Keskustelukanavia tiimin ja työpaikan sisällä ovat sähköposti, Teams ja Slack.

Varsinaisessa kehitystyössä olen tiiviisti tekemisissä tiimin kanssa ja jaamme tietoa projektin edis-
tymisestä. Työtehtäviä jaetaan tiketeillä Jiran kautta ja välillä tehtävät ovat itsenäisiä, toiset pie-
nissä ryhmissä tehtyjä. Kun tehtävät on saatu tehtyä, ne täytyy viedä versiohallinnan kautta muille
tarkasteltavaksi, jossa koodia arvostellaan ja siitä saa palautetta. Työn edistyessä saan siis pa-
lautetta muilta kehittäjiltä, jota on tärkeää kuunnella ja sisäistää kehittyäkseen työssä eteenpäin.
Muiden kehittäjien koodin tarkastelu ja kommentointi on tärkeää myös omaan kehittymiseeni.

Konsultoinnissa joudun keskustelemaan asiakkaiden, varsinkin heidän teknisten työntekijöiden
kanssa. Siinä on tärkeää osata olla selkeä ja jakaa oikeaa tietoa hyvin selitettynä. Hyvät vuorovai-
kutustaidot ovat tässä hyvin tärkeitä, sillä asiakkailla saattaa olla välillä isojaakin vaatimuksia aika-
tauluista ja tavoitteista. Keskusteluissa täytyy osata pitää puolensa ja osata välillä puhua koko tii-
min puolesta. Olen tehnyt vuosia asiakaspalvelutyötä, jossa olen kehittänyt hyvät vuorovaikutustai-
dot, enkä anna oman mielipiteeni tai persoonani liikaa vaikuttaa keskusteluun. Opiskeluni olen teh-
nyt koronan takia pääasiallisesti etänä, joten etänä käydyt keskustelut ovat arkipäivää.

Kehitystiimin sisällä on monia eri persoonia ja eri tapoja suorittaa asioita, mutta tuotekehityksessä
seuraamme selkeitä ohjeita siitä, miten asioita täytyy tehdä. Tällä välttää ainakin pahimmat riidat
koodaamistottumuksista, sillä seuraamme Microsoftin käytäntöjä siitä, kuinka asiat tulisi tehdä. En
usko, että omien vuorovaikutustaitojeni tulisi kehittyä paljoo, että pärjään työssäni. Asiakkaiden

kanssa keskustellessa mahdollisimman laajasta teknologioiden tietämisestä on apua ja siihen pyrin. Oman tiimin kesken tulen kaikkien kanssa toimeen ja osaan kuunnella ja oppia uusia asioita. Voisin kehittyä rohkeammaksi omien ideoitteni esilletuomisessa, mutta se varmasti kehittyy sen mukaan, mitä paremmaksi kehityn itse työssäni.

3 Seurantajakson raportointi viikkoanalyysiin

Seurantajakson raportointi sisältää päivittäisen läpikäynnin töistäni, sekä aina viikon lopussa viikkoanalyysin. Päiväkirjamerkintöjen on tarkoitus olla selostava ja pohtiva, kun taas viikkoanalyysi sisältää teoreettisempaa sisältöä viikolla läpikäydyistä asioista. Pyrin poimimaan viikkoanalyysiin tärkeimpinä kokemiani asioita, jotta saan niihin itselleni teoriaa tueksi ja pystyn kehittämään parhaiten työtehtävissäni.

3.1 Seurantaviikko 1

Maanantai 04.04.2022

Maanantai lähti aiheella, jota olin käsitellyt jo edellisellä viikolla. Olimme tutustunut Azuren funktioihin, koodanneet oman sovelluksen ja siirtäneet sen toimimaan Azureen. Viimeisenä aiheena siihen liittyen teimme vielä testejä sovelluksen toiminnan varmistamiseksi. Sain tehtäväksi kirjoittaa integraatiotestin, joka varmistaisi sovelluksen ja tietokannan väliltä, että tietoa pystyy hakemaan, ja että sieltä löytyisi tietty haettu tieto.

Tuleva työtehtäväni sisältää periaatteessa vain yksikkötestausta, mutta tänään teimme integraatiotestin. Testi rakennettiin kolmen A:n periaatteella Arrange/Act/Assert eli Aseta/Toimi/Vaadi. Tämä on paljon myös yksikkötestauksessa käytetty malli ja täten hyödyllinen oppia. Aseta-vaihe tarkoittaa sitä, että asetetaan mitä tietoa haetaan ja mistä sitä haetaan. Tiedon olisi voinut hakea Azuresta, mutta käytimme testaustyökalua nimeltään Mock. Mock palvelun pystyy esiintymään tietokantana ja sen voi laittamaan palauttamaan minkä vain asetetun tiedon. Täten testin ei tarvitse hakea tietoa oikeasta kannasta, eikä minun tarvitse tietää onko sitä tietoa oikeasti olemassa, mutta saan tiedon siitä, toimiiko testi itsessään.

Toimi-vaihe tarkoittaa sitä, että tietoa haetaan. Voidaan esimerkiksi hakea kappaleen nimeä soittolistalta. Itse kutsuin tekemääni sovellusta tässä vaiheessa, joka sisälsi musiikkilistoja ja olisin voinut hakea tietokannasta mitä vain kappaletta nimellä. Käytin kuitenkin Mock palvelua ja kappaleen nimi on tässä vaiheessa keksitty. Vaadi-vaiheelle löytyy Visual Studiosta oma luokka Assert. Tällä luokalla voidaan verrata tuloksen oikeellisuutta vaadittuun tulokseen ja testi joko menee läpi tai palauttaa virheen.

Olen pitänyt testausta urani kompastuskivenä ja pelkäsin tämän päivän koulutusta valmiiksi, mutta tämä opetustyyli ja tekniikat olivat varsin opettavaisia. Otan AAA tyylin käyttöön, sillä se selkeyttää testin rakentamista selvästi ja Mock-kirjasto helpottaa tekemistä entisestään. Testit ovat tärkeitä suurissa projekteissa pitääkseen koodin eheänä ja toimivana, sillä jos ongelmia tulee, ajamalla tes-

tit voidaan saada ongelmakohta selville. D365FO kehittäjänä testit tehdään pääsääntöisesti yksiköttestauksina, mutta nekin rakennetaan AAA tyylillä. Tämä jälkeen testit voidaan liittää DevOps-palveluun, josta niitä ajettaisiin automaattisesti varmistaakseen järjestelmän toiminnan.

Tiistai 05.04.2022

Tiistaina tutustuin tarkemmin Visual Studioon ja Finance and Operations ympäristön luomiseen. Minulle oli valmiiksi luotu Azureen virtuaalikone, jonne oli asennettu Visual Studio. Otin ensiksi yhteyden koneeseen Lifecycle Servicesin kautta ja säädin asetukset oikein firman käytäntöjen mukaisesti. Tärkeimpänä virtuaalikoneen automaattisen sammutuksen, ettei kone jää päälle turhaan ja laskuta ylläpidosta.

Visual Studiossa on tärkein asettaa se aukeamaan aina järjestelmänvalvojan oikeuksilla. Tämän jälkeen se on yhdistettävä versiohallintaan. Koneelle on myös ladattava ja asennetta uusin PowerShell, sekä yhdistettävä se malleja sisältävään kansioon, jotta mallit rakentuvat kunnolla. Tämän jälkeen Visual Studiosta on mentävä Dynamics 365 valikkoon ja mallit on päivitettävä, jonka jälkeen ne on laitettava rakentumaan. Mallien tulisi tulla projektin versiohallinnasta ja niiden tulisi rakentua ilman ongelmia. Tämän jälkeen FO-järjestelmä on rakennettu ja siihen pystyy ottamaan yhteyttä. Järjestelmässä kävin vielä antamassa itselleni oikeudet kirjautua sähköpostilla, jonka jälkeen sitä pääsee käyttämään selaimelta.

Virtuaalikone ja koko ympäristö on rakennettu niin sanotusti hiekkalaatikoksi, joka on minun leikkikenttäni. Saan tehdä kaikkia mahdollisia muutoksia, koodata mitä haluan ja päivittää muutoksia versiohallintaan ilman pelkoa siitä, että mikään rikkoutuisi. Se, että pääsin vihdoinkin käsiksi Finance and Operations näkymään, oli mahtavaa. Olen nähnyt aiemmin FO-näkymän, mutta nyt kun pääsin tutustumaan ympäristöön, alkoi työtehtäväni selkenemään entisestään. Visual Studio on edelleen erilainen kehitysympäristönä kuin muut käyttämäni, mutta alan saamaa siihenkin otetta.

Keskiviikko 06.04.2022

Keskiviikkona tutustuin versiohallintaan, joka FO-puolella hoidetaan Azure DevOpsin kautta. Visual Studio yhdistetään DevOpsin palvelimeen ja tämän jälkeen versiohallinta on helppoa suoraan Visual Studion kautta. Ohjelmointitöitä jaetaan Jira-tiketeillä ja luodessa Visual Studioon uutta haaraa, siihen on lisättävä aina Jira-tiketin numero. Uusin versio tulisi aina päivittää päivän päätteeksi DevOpsiin, sillä virtuaalikoneilla oleva tieto saattaa hukkuu koneen sammuaessa ja uudelleenkäynnistyksessä.

Käytin loppupäivä versiohallintaa harjoitellessa, tehden uusia päivityksiä ja päivittäen niitä DevOpsiin. Uusin asia itselleni oli pull-pyyntöjen tekeminen, joten tein sitä paljon. Versiohallinta on aina

ollut hiukan pelottava asia, sillä väärällä päivityksellä voi lamauttaa koko tuotteen toiminnan. Varsinkin työympäristössä. On siis tärkeää seurata virallisia ohjeita ja loppuillan harjoittelu omassa ympäristössä ja se toi paljon itsevarmuutta toimimiseen. Vielä myöhemmäksi hetkeksi jää tutustuminen muihin DevOpsin ominaisuuksiin. Suurimpana kiinnostaa pipeline ja testien luominen DevOpsiin. Tähän päivään opiskeltavaa oli kuitenkin jo tarpeeksi.

Torstai 07.04.2022

Tänään jatkoimme FO ympäristöjen opiskelua aiheella AOT objektit. AOT eli application object tree käsittää kaikki FO-ympäristön objektit. Objektipuuta selaamalla voi tarkastella kaikkia elementtejä, kuten luokkia, lomakkeita ja muuttujia. Harjoittelin AOT objektien lähdekoodin hakemista ja tarkastelin objektien tietokantayhteyksiä. Objekteja pystyy laajentamaan ja täten lisäämään toimintoja. Laajensin yhtä muuttujaa ja aukaisin sen tiedot. Tiedoista näin, että sen merkkijonon pituudeksi on annettu 12, joten muutin sen muotoon 20. Tallensin tiedon ja nyt muuttujaan pystyisi tallentamaan pidemmän merkkijonon.

Aloitin vapaa-ajallani käymään Microsoft Learn portaalista MB-500-kurssia (Microsoft 2022c), joka on suunnattu FO kehittäjille. Ensimmäisessä osassa käydään ympäristön perusteita läpi ja hieman objekteja. Torstain loppupäivä käytin tutustuessa objekteihin ja loin omaan järjestelmääni uuden painikkeen, josta pääsen uudelle lomakkeelle. Tein myös lomakkeen, jonka alle listattiin yksilöivällä id:llä esimerkkiyrityksen tietokoneet. Ainoastaan napin lisäämisessä täytyi käyttää laajentamista, mutta uuden lomakkeen tein alusta asti. Lisäsin muuttujat DeviceId ja CompanyId, jotka yksilöivät tiedot ja laitoin ne näkymään uudelle lomakkeelle. Nyt kun menen sovelluksessa oikeaan paikkaan, pääsen napista uudelle lomakkeelle, josta löytyy listattuna yritykset ja laitteet. Tälle lomakkeelle pystyy myös lisäämään tietoa, kuten tässä esimerkissä tietokoneita.

Perjantai 08.04.2022

Perjantaina jatkoin samaa AOT-objekteihin tutustumista. Tällä kertaa lisäsin järjestelmään käyttöoikeuksia, jotta ulkopuoliset näkisivät vain osan tiedoista. Tämän jälkeen lisäsin uudelle luokalleni oikeudet, jotta sitä pääsisi näkemään vain tietyllä sähköpostilla tai IP-osoitteella. Oikeuksien lisääminen oli yksinkertaista käyttäjähallinnassa ja kollegani pääsi käyttämään sovellustani. Jatkoin tämän jälkeen Microsoft Learn portaalin kurssia, sillä olin valmis siirtymään seuraavaan osioon ja minulla oli vapaata aikaa. Nämä sisälsivät testien tekemistä ja raportointityökaluja FO:n sisällä.

Loppupäivästä meillä oli yrityksen kuukausittainen kokous, jossa käytiin läpi työpaikan asioita. Koen nämä todella virkistävinä ja tykkään johdon avoimuudesta kertoa yrityksen asioista työntekijöille. Nykyajan etäkäytänteet ovat tehneet tästä helpompia, sillä tähänkin kokoukseen liittyi ihmisiä ympäri Eurooppaa. Itselle etäajan koulu ja työskentely on hämärtänyt omaa tekemisen tasoa.

Sitä luulee itse tekevänsä liian vähän ja tekeekin usein aivan liikaa. Tiimissäni on joka toinen päivä tiimikokouksia, sekä viikoittaisia ja kuukausittaisia kokouksia työpaikalla. Nämä kokoukset palauttavat oman käsitykseni omasta osaamisestani.

Viikkoanalyysi

Asetin tavoitteeksi tälle viikolle sen, että saan hyvän kuvan varsinaisesta työtehtävästäni ja pääsen käsiksi niihin sovelluksiin, jotka ovat työni kannalta tärkeitä. Olin kuullut kaikista näistä aiemmin, joitakin hieman käyttänyt, mutta nyt halusin tarkemman kuvan. Onnistuin viikon aikana saamaan hyvän kokonaiskuvan omasta työstäni ja kehitysalustasta. Se toi todella paljon varmuutta siitä, että opin nämä asiat ja siitä, että tulen suoriutumaan työstäni hyvin. Microsoftin viralliset ohjeet ja työpaikan omat ohjeet vaikuttavat varsin riittävältä siihen, että saan sieltä tarvittaessa apua ongelmiin. Toki minulla on myös mentori ja ihmisiä ympärilläni, jolta voin aina kysyä.

Tärkein D365FO ympäristöstä käsitellyin aihe oli AOT objektit. AOT objektit ovat kuin järjestelmän tietokantaobjektit ja täten kehittäminen vaatii näiden hyvää tietämystä. AOT objekteja pystyy jatkamaan ja täten niiden tietoja pystyy muokkaamaan tai lisäämään. AOT objekteja ovat esimerkiksi lomakkeet, taulut, elementit ja muuttujat. AOT taulusta näkee myös kaikki selitykset ja lähdekoodit objekteille. AOT objektien ja niiden käytön oikeellisuuden Microsoft tarkistaa automaattisella best practices työkalullaan. Best practices antaa ohjelmoijalle neuvoa siitä, miten koodista saa eheämmän ja toimivamman, jolloin se pysyy toimivampana jatkossakin. (Microsoft 2018a).

Testaus on hyvin tärkeää sovellusten oikean toiminnan varmistamiselle. Yleensä testauksista tehdään automaattisia, jolloin ne suorittavat itsensä aamuisin. Tällöin kehittäjät tietävät, että järjestelmässä ei ole valmiiksi virheitä, kun he lähtevät tekemään uutta. Testausta on kolmea eri tasoista: End-to-end-, integraatio- ja yksikkötestaus. End-to-end on harvimmiten tehtyä, sillä se vaatii eniten kapasiteettia, integraatiotestaus on suositelluin muuttuvissa ympäristöissä ja yksikkötestaus on helpoin, jolla varmistetaan sovelluksen sisäinen toiminta. (Codepipes 2018). Meidän työssämme yksikkötestaus on yleisin, sillä kyseessä on sisäinen järjestelmä, joka ei ole paljon tekemisissä ulkoisiin järjestelmiin. On vain varmistettava tiedon eheys. Testauksessa käytetty AAA eli Arrange-Act-Assert tyyli on selkeä ja käytännöllinen tapa myös FO-ympäristössä. (Telerik 2022).

Tämä oli ensimmäinen seurantaviikkoni opinnäytetyössä, mutta varsinaisessa työssäni se oli jo neljäs. Pidän sitä varsin hyvänä, että olen tähän mennessä oppinut laajasti asioita, sillä organisaatiomme sisällä tehtäviä ja toimintatapoja on monia erilaisia. Päästessäni tutustumaan omaan tiimiini ja tehtävääni, osaan kuitenkin sanoa, että olen valinnut oikean tehtävän. Tai vaihtoehtoisesti, minut on palkattu oikeaan tehtävään. Tehtäväni on rajatussa ympäristössä koodaaminen, mutta tämä ympäristö kuitenkin tarjoaa todella paljon opeteltavaa ja tehtävää. Minun ei tarvitse

luoda kaikkea tyhjästä, mutta pääsen kuitenkin kehittämään uutta koodia. Tutustuimme aiemmin Azuren low-code ja no-code ympäristöihin, ja siinä missä ne ovat mielenkiintoisia ja käytännöllisiä, en näe niiden kehittävän omaa ammattiosaamistani tarpeeksi.

Olen työviikon jälkeen vielä käynyt Microsoft Learnin kurssia eteenpäin ja tutustunut tuleviin aiheisiin. Ensi viikon tavoitteiksi asetan itselleni X++ koodikielen alkeiden sisäistämisen. Microsoftin kurssi ja työpaikan opetus tulee opettamaan paljon, mutta olen jo lukenut aiheesta muistakin lähteistä. Sen lisäksi aio kerrata tämän viikon aiheita vielä uudestaan ja harjoitella niitä omassa ympäristössäni. Tämä kaikki tietysti auttaa työssäni, mutta myös sisäistämään Microsoftin virallisten ohjeiden mukaiset toimintatavat, joka taas helpottaa tulevan sertifikaatin suorittamista. Ensi viikkoa odotan innolla!

3.2 Seurantaviikko 2

Maanantai 11.4.2022

Tänään jatkoin vielä AOT-objektien läpikäyntiä ja kertosin viime viikon asioita. Viikonlopuksi minulla oli Microsoftin materiaaleja läpikäytäväksi ja ohjaaja oli järjestänyt pienen pistokokeen aiheista. Kysymyksiä oli muun muassa siitä, mitä Microsoftin työkaluja käytetään D365FO raportoinnissa, millä hakusanaa metadatahaussa käytetään, kun haetaan koodia, mikä on task recorder ja mitä työtilaa käytetään projekteissa: Julkista vai yksityistä. Osasin vastata kaikkeen vähintään suuntaa antavasti, eli kaikista asioista oli jotain jäänyt mieleen. Se vahvisti tunnetta, että osaan aiheet tarpeeksi, jotta pystyn itsenäiseen toimintaan.

Tämän jälkeen loin FO-ympäristössä konfiguraatioavaimen. Tämä avulla pystyn järjestelmän ylläpitäjänä hallinnoimaan käyttäjien oikeuksia järjestelmässä. Avaimen luominen on itsessään helppoa. Tämän jälkeen se tulee linkittää display menu itemiin. Tämän jälkeen on luotava valmiit käyttöoikeus objektit. Nyt kun ohjelmiston käynnistää ylläpitäjänä, käyttöliittymästä pystyy käydä antamassa käyttäjille itse luotuja käyttöoikeuksia. Tämän jälkeen harjoittelin vielä tiettyjen tietojen hakemista ja tietojen tuomista esimerkiksi Exceliin. Tämä oli helppoa käyttöjärjestelmästä, kun vain löysi oikean paikan.

Loppupäivän tutustuin toisen ohjaajan kanssa FO:n käyttönäkymään ja siihen, kuinka siellä liikutaan ja haetaan tietoa. FO:n aloitusikkuna näyttää heti montaa kymmentä eri valintaikkunaa ja valikkoa painamalla niitä aukeaa vielä useampia. Se on siis valtavan oloinen järjestelmä ja alkuun tuntuu siltä, ettei sieltä osaa edes lähteä hakemaan tietoa. Tämän takia oli varsin mielenkiintoista tutustua järjestelmään ammattilaisen kanssa. Koulutus oli vain tunnin pituinen eli siinä ei päästy

kovin syväälle, mutta sain hyvää tietoa. Tärkeimpänä jäi mieleen se, että osaan katsoa järjestelmästä jokaisen elementin ja tiedon lähteen, eli pääsen siihen nopeasti käsiksi koodin kautta. Osaan katsoa paremmin mistä tieto tulee.

Tiistai 12.4.2022

Tiistaina pääsin vihdoinkin käsiksi D365FO:n viralliseen koodauskieleen eli X++:aan. X++ on paljolti C#:iin pohjautuva koodikieli. Minulla ei ole kokemusta näistä kielistä, mutta olen koodannut paljon niin frontend, kuin backend-kielillä. X++ on fullstack objektikieli, joten aiemmasta laajasta osaamisestani on hyötyä. Emme käyneet teoriaa läpi, vaan lähdimme suoraan tekemään esimerkkikoodia. FO ympäristö on alueena rajattu ja täten koodin kirjoittaminen on myös rajattua. Riippuvuuksia ei tarvitse hakea netistä, mutta järjestelmään on linkitettävä eri paketteja käytettävien taulujen mukaan. Ensiksi FO järjestelmään täytyy luoda uusi runnable class, josta uusi koodimme on ajettavissa.

Ensimmäisenä yllätyksenä on se, että X++ ei ole case sensitive kieleltään, eli kaiken voi kirjoittaa pienellä, isolla tai yhdistelemällä, kunhan sana on oikea. Toiseksi kaikki kirjoitettu koodi menee Microsoftin automaattisen tarkkailun kautta, joka varmistaa, että olet tehnyt kaiken parhaiden käytäntöjen mukaan. Harjoituksessamme käynnistimme FO-ympäristön aina uudelleen, kun halusimme nähdä muutoksia koodissa. Se teki etenemisestä hiukan hidasta, sillä järjestelmän käynnistäminen ja muutosten latautuminen on melko hidasta. Varsinaisessa asiakastyössä FO-ympäristö on aina auki selaimella ja muutoksia pystyy tarkastelemaan aina kun uudet muutokset tallennetaan ja ne laitetaan rakentumaan.

Harjoittelimme aluksi ilmoitusten luomista ja niihin tietojen lisäämistä. Ilmoituksia on info, error ja warning. Muuttujat ovat esimerkiksi tyypilliset string ja integer, mutta sen lisäksi on esimerkiksi liukuluku real. Kun tieto on muuttujissa, sitä on helppo kutsua. X++ erikoisuutena on tietokantakyselyt. Haimme järjestelmän asiakastaulusta yritysten nimiä ja niiden numeroita. X++:lla pystyy kirjoittamaan myös SQL lausekkeita ja toteuttamaan täydet CRUD-toiminnot. Eli lisäämään, muokkaamaan, poistamaan ja päivittämään tietoa. Harjoittelimme vielä tiedon päivittämistä. Siinä ei it-sessään ollut hankaluuksia, mutta tärkeimpänä on muistaa merkitä mistä lause alkaa ja loppuu, jottei se jää päivittämään loputtomasti.

Keskiviikko 13.4.2022

Jatkoimme X++ kielen opiskelua ja CRUD-toimintojen seuraavia vaiheita eli tietojen lisäystä ja poistamista. Ne toimivat samalla periaatteella, eli tieto haetaan SQL-lausekkeella ja toteutetaan omalla toiminnolla. Tapahtuma täytyy muistaa jälleen kääriä aloituksen ja lopetuksen sisään. Kun lausekkeet on luotu, ohjelmisto on käynnistettävä ja toiminnon toiminnallisuus käytävä toteamassa.

Tämän jälkeen toteutin virheidenkäsittelyn ohjelmakoodiin. Tällä hetkellä olin tilanteessa, jossa jokaisella käynnistyksellä ohjelmisto yrittää lisätä tai poistaa samaa tietoa, joten tein virheidenkäsittelyn. FO ympäristössä käytetään yleisimmin Try...Catch-lausetta. Eli tietoa yritetään lisätä, mutta koska se ei onnistu, järjestelmä palauttaa virheviestin. Useammalla Catch-lausekkeella voi laittaa eri viestejä ja finally-komennolla voi määritellä mitä palautuu joka tapauksessa. Tässä ei ollut itselleni paljoa uutta, mutta hyvää kertausta uudessa järjestelmässä.

Iltapäivästä jatkoin opiskelua FO-käyttöjärjestelmän kanssa ja tällä kertaa katsoimme syvemmälle, miten esimerkiksi yrityksen myyntitietoja haetaan ja kuinka järjestelmästä voi tulostaa raportteja tai kuitteja. D365FO järjestelmässä on demoasiakas nimeltään USMF. Yritykselle on lisätty demodataa, jota voi tarkkailla ja muuttaa haluamallaan tavalla. Loimme kuvitteellista myyntiä yritykselle, tulostimme kuitteja myynnistä ja teimme yhteenvetoja myynnistä. Oli todella mielenkiintoista nähdä miten konsultit ja asiakkaat tulevat järjestelmää käyttämään ja se tietenkin antaa perspektiiviä myös omaan osaamiseen.

Torstai 14.4.2022

Olimme käyneet kaikki koulutukseen liittyvät asiat Visual Studiosta, X++ kielestä ja FO-ympäristöstä, joten enempää tehtävää uutta tehtävää ei ollut. FO-ympäristö kuitenkin rakentuu best practices käytännöille ja viikko tekemisen jälkeen herjoja oli yli kaksikymmentä. On tietysti suositeltavaa, että herjoja ei olisi yhtään, joten aloin tutkimaan mistä ne johtuvat ja lähdin korjaamaan niitä. Yleisimmät virheet johtuivat siitä, että tietojen viittauksia puuttui joistain elementeistä. Nämä elementit täytyi liittää oikeisiin tauluihin ja virheet poistuivat. Toinen yleinen oli se, että kaikki muuttujat tuli lisätä etikettien taakse. FO-järjestelmässä luodaan etikettejä, josta muuttujia voidaan kutsua ja tämä on Microsoftin parhaiden käytäntöjen mukaista. Tämä jälkeen oli vain pieniä virheitä, kuten väärin heittomerkkien käyttö. Loppupäiväni meni näitä korjatessa ja siinä, että kävin Microsoft learn materiaalia taas eteenpäin.

Perjantai 15.4.2022

Pitkäperjantai

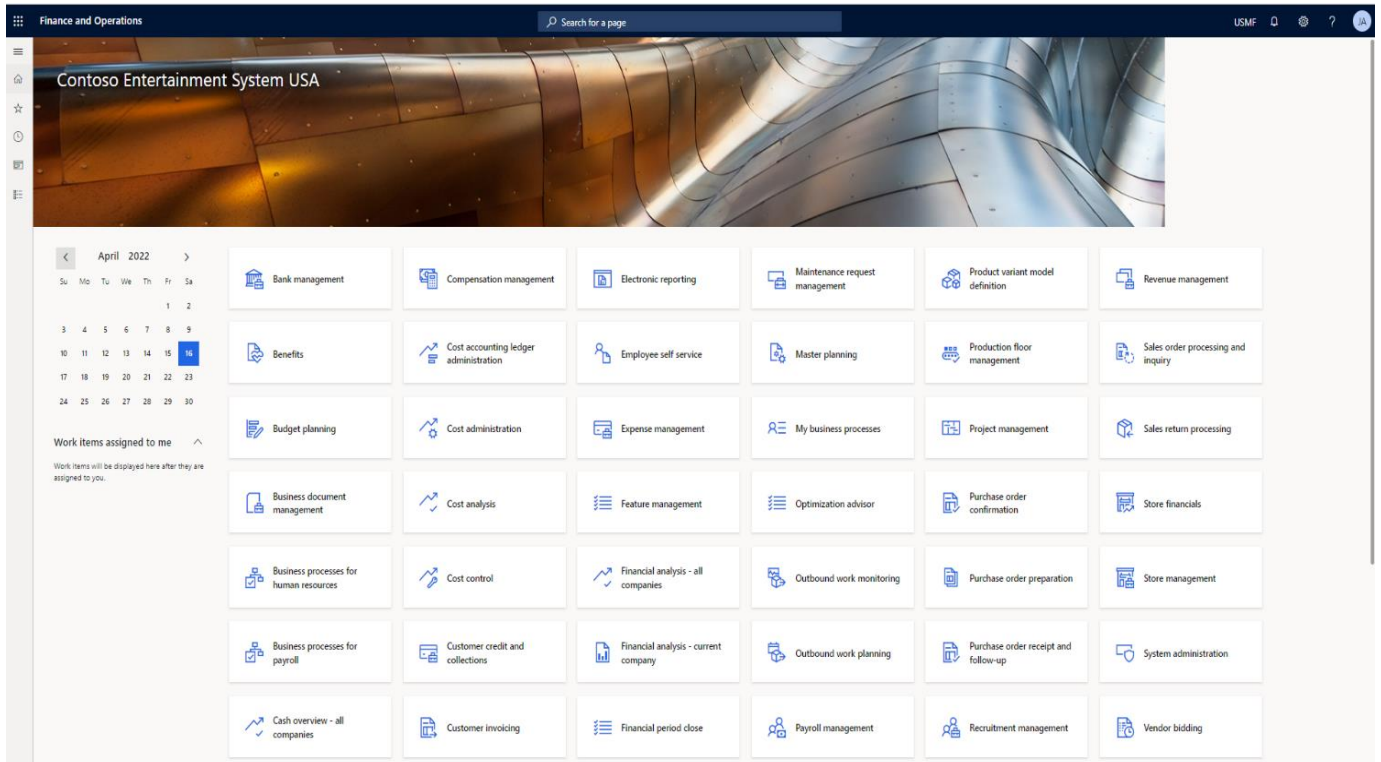
Viikkoanalyysi

Tavoitteeni oli X++ kielen perusteiden oppiminen ja onnistuin siinä suhteellisen hyvin. Alkuperäiset odotukseni siitä, millainen viikko olisi, olivat erilaiset. Kuvittelin sen olevan paljon teknisempää ja kovempaa koodaamista, mutta se olikin enemmän ympäristössä toimimisesta. Opin paljon enemmän käyttöjärjestelmästä ja siitä, kuinka siellä haetaan tietoa koodaamista varten. Tämä tiedonhaku on ollut alussa vaikein asia itselleni sisäistää, sillä tietorakenne on niin laaja. Koodaaminen

X++ kielellä on hyvin samantyylistä kuin useissa muissakin kielissä. Siinä on paljon samat muuttujat ja samoja funktioita, mutta niiden käyttö on entistä tarkempaa, kun kaiken tekee parhaiden käytäntöjen mukaan. Täten liian tekninen läpikäynti ei ollut tarvittavaa, vaan mennyt viikko oli juuri sopivaa jatkoa aiempaan.

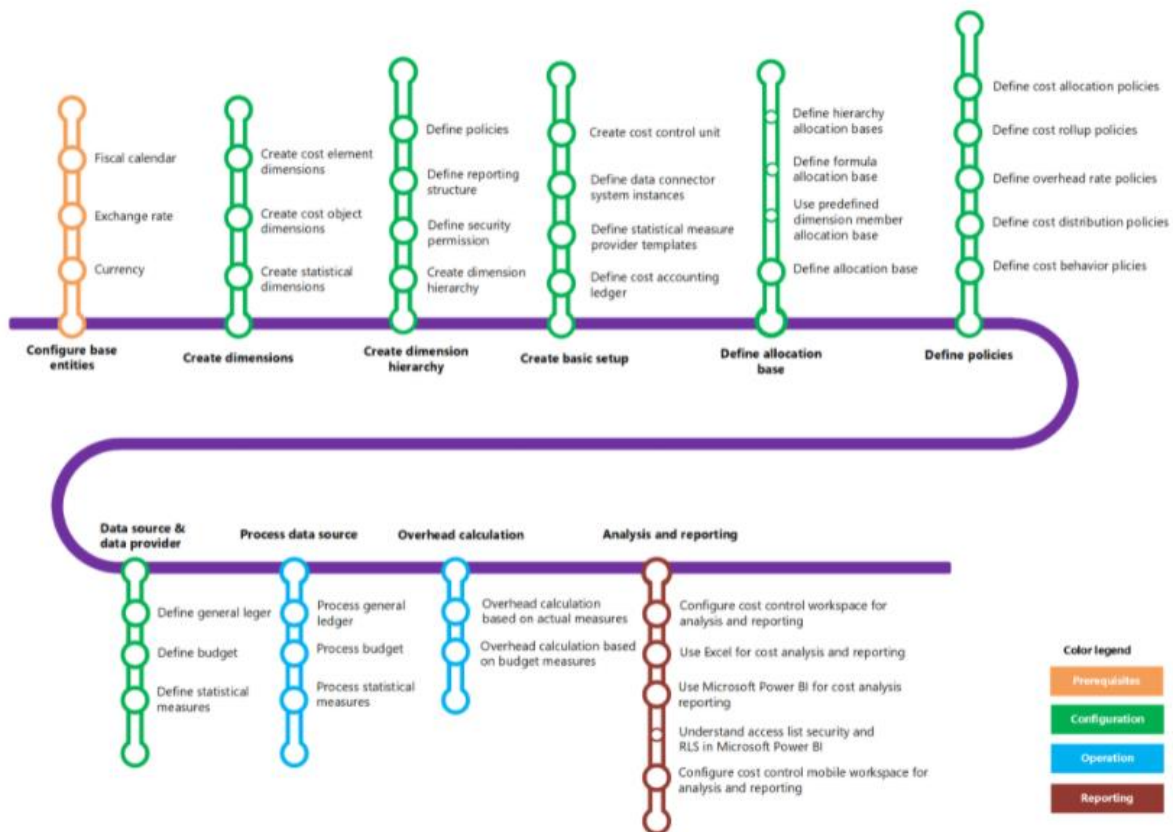
X++:ssa kuitenkin oli joitakin eroja esimerkiksi itselleni tutumpiin kieliin, jotka vaativat totuttelua. Ensimmäisenä ilmeisin eli se, että X++ ei ole case sensitive kieli eli kirjainten koolla ei ole merkitystä muuttujissa. Muuttujien nimet kirjoitetaan kuitenkin perinteisesti camel casingilla eli jokainen uusi sana muuttujien nimessä alkaa isolla alkukirjaimella. Muita yksinkertaisia eroja ovat if-else lausekkeen toiminta, merkkijonot ja viestien kirjoittaminen. X++ if-else lausekkeeseen voi laittaa minkä tahansa muuttujan, jonka se voi automaattisesti kääntää totuusarvomuttujaksi. Merkkijonot voi kirjoittaa joko ” tai ’ heittomerkeillä. Viestit voi kirjoittaa joko info(), warning() tai error() muodoissa, kun esimerkiksi C#:ssa on Console.Out.WriteLine() ja Javassa System.out.println(). Myös tietyt avainsanat ja funktioiden toiminnan eroavat toisistaan. Näkisin kuitenkin, että suurin ero C# verrattuna on toimintaympäristö ja käyttötarkoitus. (Microsoft 2021).

Dynamics 365 Finance and Operations on käyttämäni ohjelmisto. FO on yritykselle tärkeä työkalu, jolla voi tarkkailla yrityksen välistä kauppaa, kauppojen oikeudenmukaisuutta, toimitusketjua ja tuotteiden valmistusta. Se myös helpottaa laskujen tekemisessä, lähettämisessä ja tarkkailussa. (Fox 2022). FO on suunnattu isompien ja monimutkaisempien yritysten käyttötarkoituksiin, esimerkiksi verrattuna Microsoftin Business Centraliin.



Kuva 2. Esimerkinäkymä D365FO:sta

Kuva 2 esittää esimerkinäkymän FO:n aloitussivusta. Asiakkaalle valinnoista näkyisi vain murto-osa, mutta ohjelmoijana ja pääkäyttäjänä näen kaiken. Järjestelmän esimerkkirytyksenä on Contoso Entertainment System USA, jolla on esimerkkidataa. Microsoft tarjoaa omassa materiaalis- saan oppimiskartan (kuva 3) maksun kirjanpitoon, joka kiteyttää parhaiten harjoittelemani asiat. Maksun tekemisen voi jakaa neljään päävaiheeseen, joita ovat alkutiedot, konfiguraatio, operaatio ja raportointi. (Microsoft 2022d).



Kuva 3. Raportin tekeminen (Microsoft 2022d)

Jatkoin pitkin viikkoa Microsoft Learnin materiaalin lukua. Käyn FO kurssi, joka johtaa lopulta sertifiikaattiin. Se on hyvin niputtanut viikolla käsiteltäviä aiheita yhteen tuoden yksityiskohtaista tietoa mukaan.

Tässä viikossa oli kova kiireen tuntu ja tuntui, että tehtävien tekeminen jatkui aina iltaan asti. Lopulta asiat ovat kuitenkin melko selkeitä ja päiväkirjaa kirjoittaessa asiat tuntuivat jo paljon simppeleimmiltä. Ensi viikon tavoitteeksi otan DevOpsiin syventymisen. Kouluttajan tulisi antaa myös tehtäviä omalle ajalle tulevaisuudelle viikoksi, sekä alan käymään meidän osastomme aamutapaamisissa. Tätä kautta näen mitä muut tekevät, pääsen esittelemään itseni ja saatan saada tehtäviä, jotta pääsen alkuun virallisesti työnteossa. Harjoittelu alkaa olemaan viiden työviikon jälkeen lopussa ja varsinainen työ edessä. Alun hermostuminen on kääntynyt vahvasti sen puolelle, että odotan varsinainen asiakasprojektien alkua.

3.3 Seurantaviikko 3

Maanantai 18.4.2022

Pääsiäinen

Tiistai 19.4.2022

Pääsiäisen jälkeen kävimme kouluttajan kanssa toimitusprosesseja läpi eli Azure DevOpsissa projektin sinne päivittämistä, rakentamista ja testausta. Kun vien uuden version versiohallintaan, on siihen sisällytettävä viesti tehdyistä muutoksista ja nimettävä vähintään yksi ihminen, joka käy arvioimassa ja hyväksymässä muutokset. Viesteissä pitää muistaa käytäntönä kirjoittaa viestit nyky-muodossa, ei menneessä. Vedimme versiohallinnasta muiden päivityksiä omaan projektiimme. Tässä on joskus häiriötä binäärimuotojen kanssa ja muutamia Microsoftin automaattipalveluja täytyi laittaa pois päältä sen ajaksi. Loimme päällekkäisyyksiä koodeissa ja selvitimme, kuinka niiden selvittäminen onnistuu.

Kun päivitys oli oikein laitettu järjestelmään, se näkyi Pipelinessa, josta se menisi automaattisesti rakennettavaksi. Harjoituksen vuoksi laitoimme uuden version rakentumaan heti, jotta varmistuimme toiminnasta. DevOpsin Pipeline sisältää valmiiksi yaml-tiedoston, jonka mukaan sovellus rakentuu. Microsoftilta vaadittavat tiedot rakentamiseen tulevat suoraan zip-tiedostona. Pipeline voisi muokata haluamakseen valitsemalla muokattu Pipeline, mutta se ei ole yleensä tarpeellista. Tarkistimme vielä version oikeellisen toiminnan Lifecycles Servicestä, joka tarkkailee ympäristön toimintaa ja josta rakennettuun järjestelmään pystyy yhdistämään.

Päivän aiheena ei ollut testaus, mutta DevOpsiin pystyy rakentamaan automaattisia testaus suunnitelmia, jotka voivat ajaa itsensä esimerkiksi joka aamu. Itse testi on kirjoitettava koodiin, mutta DevOpsista sitä voi kutsua. Regression Suite Automation-työkalulla voi rakentaa itse testausjärjestelmän, jossa automaatio seuraa hiiren klikkauksia ja rakentaa testin itse. Minua kuitenkin kiinnostaa enemmän itse rakennettu testi ja sen saa tehtyä Visual Studioon omana luokkana. Aloin rakentamaan testiä AAA-tyylillä, sain sen valmiiksi, mutta en ehtinyt rakentamaan testitapausta DevOpsiin.

Keskiviikko 20.4.2022

Osallistuin itselleni ensimmäiseen tiimipalaveriin, joita meidän osastollamme pidetään joka toinen päivä. Palaveriin osallistuivat kaikki osastomme työntekijät ja läpikäyntiä johti esimiehemme. Hän kyseli minun tilanteistani, muiden etenemisestä ja yleiskuvasta. Vielä tällä hetkellä ei ollut suoraan projektia, johon minua voisi liittää, mutta se tilanne tulee aktiivisesti elämään. Tärkeintä itselleni oli

huomata, millaista keskustelu muiden välillä oli. Kaikki pystyivät kertomaan omasta tilanteestaan ja he pyysivät apua toisiltaan. Ihmiset todella helposti kysyivät toisiltaan, että voisiko joku tietty henkilö tulla myöhemmin Teams tapaamiseen katsomaan koodia ja vertailemaan tilannetta. Tällainen avoimuus ja keveys toi todella paljon helpotusta siitä, että en tule olemaan yksin työssä.

Kertasimme päivästä oman koulutukseni tilannetta ja kouluttaja oli varautunut kertauskysymyksillä. Tällä kertaa ne käsittelivät X++ ominaisuuksia, luokkia ja käytäntöjä. Kouluttajallani ei ollut tällä hetkellä uutta viikkotehtävää minulle, vaan loppuaika oli varattu omatoimiseen materiaalin läpikäymiseen. Microsoft on laskenut osioiden läpikäymiseen hyvin reilusti aikaa ja kouluttajani meni heidän laskujen mukaan. Loppuviikosta minun olisi käytävä materiaalit loppuun. Minulla oli vielä kolme osiota läpikäymättä ja näiden yhteenlaskettu aika oli noin 20 tuntia. Kun osiot on käyty läpi, minulla on ainakin kuva koko laajuudesta ja osaisin myöhemmin palata oikeisiin kohtiin tiedonhaku varten.

Torstai 21.4.2022

Tänään kokonainen päivä oli omaehtoista tekemistä sertifikaattikoulutuksen ja tehtävän kanssa. Seuraavana aiheena sertifikaattikoulutuksessa oli FO-sovelluksen laajentaminen. Aloin käymään tätä läpi aamupäivästä ja koulutukseen oli varattu neljä tuntia. Materiaali oli melko hyvin samaa, kuin se mitä minun täytyi viikkotehtävässäni tehdä. Se käsitteli valmiiden elementtien laajentamista ja sisälsi harjoituksia, jossa laajennettiin olemassa olevia lomakkeita. Microsoftin materiaali sisältää labroja, jotka aukeavat selaimella ja pystyt tekemään harjoitteen. Jotkin Microsoftin harjoitteista toimivat, mutta jotkin eivät auenneet selaimella ollenkaan. Käytin siis loppupäivästä aikaa, että tein nämä harjoitteet omassa ympäristössäni. Harjoitteet ovat hyvin yksinkertaisia ohjeiden seuraamisesta, mutta ne näyttävät usein nopeimman ja helpoimman keinon asioiden suorittamiseen. Koen siis tärkeänä sen, että teen niistä jokaisen itsenäisesti.

Perjantai 22.4.2022

Tänään aamusta osallistuin ensiksi tiimikokoukseen. Tiimikokous noudatteli samaa kaavaa kuin aiemmin ja työntekijät kysyivät paljon apua toisiltaan. Minua ei erityisesti huomioitu, vaan saan jatkaa harjoitustehtävän parissa. Itsellenikään ei ollut kysymyksiä. Jatkoin sertifikaattialueen läpikäyntiä ja Microsoftin harjoitteiden tekemistä. Kouluttajanani oli varannut aikaa tähän, eikä minulla ollut muuta tekemistä, joten jatkoin yksin opiskelua. Loppuviikko on ollut hiukan hiljaisempaa, mutta en ole kokenut sitä huonona asiana. Olen saanut tehdä omia harjoitteita testiympäristössä, kokenut useita virheitä ja olen testaillut versiohallinnan toimintoja. Mitä enemmän olen saanut vapautta tehdä asioita näissä ympäristöissä, sitä enemmän sen on poistanut pelkoa toiminnoista.

Viikko päättyi kuukausittaiseen yrityksen kokoukseen, jossa esiteltiin tuloksia, tulevaisuutta ja tavoitteita. Koen nämä aina todella piristäväksi ja minulle tulee olo, että olen osa suurempaa kokonaisuutta. Se on tärkeää, varsinkin näin etätyöaikoina.

Viikkoanalyysi

Tämä viikko oli jo selkeästi itsenäisempi kuin aiemmat. Aiemmin olen seurannut ohjeita ja koulutusta, kun tällä viikolla tekeminen siirtyi selkeästi omatoimisempaan. Aluksi se on vaikeaa ja tulee olo, että ei osaa mitään, mutta tekemällä sekin lähti liikkeelle. Kouluttajan sanoin ja omalla kokemuksella, tiedonhaku on yksi tärkeimmistä taidoista millä selviää tällä alalla.

D365FO:n ja X++:n vahvuus on juuri se, että niiden kehitysympäristö ja siellä olevat kehykset ovat valmiita. Tämä säästää aikaa suuresti esimerkiksi lomakkeiden tekemisessä, sillä lomakkeiden sisältöä voi rakentaa valmiista painikkeista, suodattimista, ruudukoista ja datalähteistä. Tietokannat näiden välille syntyvät automaattisesti, joten tietokantojen rakentaminen jää kokonaan pois, ainoastaan lisäys, poisto ja muokkaus on jäljellä. Kun lomake on luotu, sen voi käynnistää, jolloin Visual Studio debuggaa sovelluksen ja aukaisee sen selaimelle. Selaimelta sovelluksen kaikkia elementtejä ja niiden lähteitä pystyy tutkimaan eli elementtien paikka koodissa on helposti paikannettavissa. (Ramer 2021).

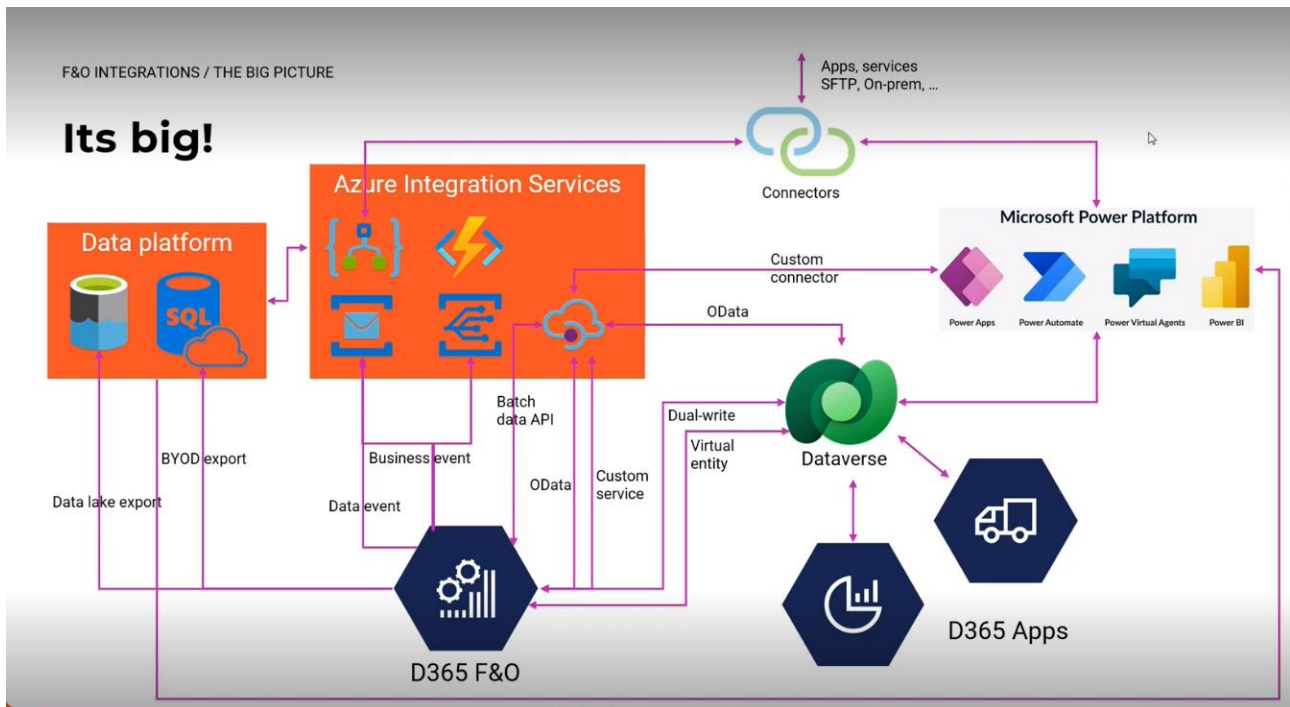
Osallistuin tällä viikolla ensimmäisiä kertoja tiimimme aamupalavereihin. Nämä järjestetään etänä siksi, että työntekijöistä suurin osa työskentelee etänä. Mäntynevan (2018) kirjassa Hallittu Projekti, annetaan ohjeita, kuinka virtuaalitiimin toimintaa voidaan ja tulisi tukea. Projektin hallinnon ja raportointisuhteiden tulisi olla selkeitä kaikille ja jäsenten nimien, valokuvien ja yhteystietojen tulisi olla kaikkien käytettävissä. Jäsenten vastuualueiden tulee olla selkeät ja työtavat, sekä käytännöt yhtenäisiä. Raportoinnin täytyy olla selkeää ja raportoinnin sisältö ja tiheys määriteltyä. (Mäntyneva 2018, 12–13).

Tämän viikon perusteella nämä toteutuvat hyvin. Jokaisen tapaamisen on aloittanut tiimin esimies ja kaikkien kuvat, nimet ja yhteystiedot ovat näkyvillä. Aluksi minut esiteltiin ja muut saivat esitellä itsensä ja toimenkuvansa minulle. Yhteystiedot löytyvät yrityksen sisäisestä verkosta. Työnjako on toteutettu Jira-tiketeillä ja esimies esitteli näkymää, josta näkee jokaisen tehtävän kullakin hetkellä. Tämän avulla on helppoa tietää mitä kukin tekee ja seurata heidän edistymistään. Työtavoista ja käytännöistä esimerkiksi koodin osalta on selkeät ohjeet sisäisessä verkossa ja myös se on sovittu, että päivän työt päivitetään aina versiohallintaan ja pääsy projekteihin pidetään julkisena. Esimies johti tapaamisia selkeästi, jakoi puheenvuoroja ja tiimin jäsenet raportoivat edistymisestä selkeästi. Tapaamiset järjestetään joka toinen päivä aamuisin, joka on mielestäni varsin sopiva väli.

3.4 Seurantaviikko 4

Maanantai 25.4.2022

D365FO järjestelmään tuodaan ja sieltä vietään tietoja myös rajapintojen kautta. Integraatioiden tekeminen itsessään on toisen tiimin erikoisosaamista, mutta kävimme tänään kouluttajan kanssa läpi peruseriaatteita sen toiminnasta. Myöhemmin viikolla tekisimme yhden esimerkin rajapinnasta. Tänään käytävä osuus oli vain nopea katsaus aiheeseen, sillä se on kokonaisuudessaan hyvin laaja kokonaisuus. Otin kuvan (kuva 4) esityksestä, jossa on kouluttajani tekemä havainnollistava selonteko integraatioympäristöstä.



Kuva 4. Azuren integraatioympäristö

Kuviota ja aihetta on helpoin lähestyä tutkimalla yhtä pistettä, joka on minun tapauksessani D365F&O. Data varastoidaan data-alustaan, joka sijaitsee Azuressa. Tämä data on linkitetty D365FO sovellukseen ja se data kulkee näiden välillä jatkuvasti. D365FO:sta eteenpäin data kulkee dataverseen. Tarkennettuna meidän tapauksessamme, se kulkee OData-muodossa, joka tekee siitä helpommin käsiteltävää ja ymmärrettävää eteenpäin. Dataversestä tietoa voi käyttää moneen muuhun sovellukseen, kuten muihin D365 sovelluksiin ja Microsoftin Power Platform sovelluksiin. Tämä on yksinkertaistettuna se, mihin minä tulen sitä eniten käyttämään, mutta kuten kuvista näkee, integraation tekemiselle on monia muitakin tapoja.

Iltapäivästä kävin läpi testisovellukseni toimintaa. Sain viime viikolla päätökseen kouluttajan antamat harjoitukset, mutta nyt tarkistimme koodin toimivuutta ja ratkaisun pätevyyttä. Ratkaisu itsessään oli hyväksyttävä ja toimiva, mutta koodia tarkastelemalla löysimme kohtia, josta sitä voisi parantaa. Olin rakentanut yhden kutsun käyttämällä kolmea eri funktiota, kun sen pystyi tiivistämään yhteen. Tämä tiivistäminen nopeuttaa toimivuutta ja varmistaa sen toiminnan pitkällä tähtäimellä. Myös koodin seuranta ja debuggaus helpottuu, kun prosessi etenee rivi riviltä, eikä hypi funktiosta toiseen. Jäin parantelemaan koodin toimintaa, jotta se olisi valmis laitettavaksi versiohallintaan. Versiohallinnan harjoittelu on kuitenkin toinen asia, joten jätän sen päivittämisen myöhemmäksi.

Tiistai 26.4.2022

Tänään jatkoimme integraation opiskelua ja lähdimme yhdistämään Azuren tietokantaa ja omaa D365FO järjestelmää. FO:n käyttöliittymään piti ensiksi lisätä integraation tiedot ja oikeudet. Hain tiedot Azuresta ja annoin niille käyttöoikeudeksi "admin" eli integraatiolla oli täydet oikeudet lukea, lisätä, muokata ja poistaa tietoa. Tämä jälkeen tietokanta on perustettava Azureen. Azuressa minulla on oma säiliö, johon perustin uuden tietokannan serverin ja sinne uuden tietokannan. Tietokantaa luodessa täytyi muistaa nimetä se samannimiseksi kuin mitä se oli FO:n puolella. Annoin siis sen nimeksi "SALogin".

Azuresta täytyi lähettää kutsu FO:iin testatakseen yhteyttä. Aluksi meillä oli ongelmia yhteyden saamisessa ja ongelmaksi osoittautui väärä tunnus. Kopioidessa pitkää tunnusta, sinne oli jäänyt loppuun vahingossa heittomerkki. Tämän korjaannuttua yhteys oli luotu ja seuraava askel olisi ulostulevan datan luominen FO:iin. FO:ssa on oma toiminto nimeltään "Data export services". Sieltä pystyy määrittelemään mitä tietoja haluaa viedä, mihin ja kuinka usein. Valitsimme esimerkiksi dataa, laitoimme sen Azuren tietokantaan ja aikaväliksi viikon. Nyt uusin data menee Azureen aina viikon välein, josta se on muiden sovellusten käytössä. Prosessi kuulostaa suoraviivaiselta, mutta saimme käytettyä asetusten laittamiseen ja toiminnan varmistamiseen kokonaisen päivän. Integraatorajapintojen teko muihin sovelluksiin ei ole ensisijaisesti työtäni, mutta Azuren kanssa työskenteleminen ja tämän toiminnallisuuden näkeminen oli mielenkiintoista.

Keskiviikko 27.4.2022

Keskiviikko alkaa suunnitelluilla tiimipalavereilla. Palaveri eteni normaalisti ja puhuimme, että tällä hetkellä minun on tärkein keskittyä omiin harjoituksiini testiympäristössä, sekä sertifiointikoulutuksen jatkamiseen.

Sain kouluttajaltani seuraavan tehtävän, jota tekisin tämän viikon. Minun täytyisi tehdä suodatin, joka suodattaisi vain tietyn yrityksen laitteet näkymääni. Esimerkkiyrityksiä on noin kaksikymmentä ja olen aiemmin tehnyt taulun, johon voin lisätä näille yrityksille laitteita. Nyt kun valitset yrityksen

nimen ja paina laitteet välilehdelle, tulisi tulla näkymään vain sen yrityksen laitteet, kun aiemmin siinä on näkynyt kaikki laitteet kaikilta yrityksiltä. Vihjeeksi sain vain muutamia esimerkkejä, josta saattaisin löytää tietoa, mutta muuten lähdin suorittamaan tehtävää itse. Lähdin liikkeelle tiedonhausta netistä ja yritin etsiä valmista luokkaa, jossa vastaava olisi toteutettuna. Tiesin kuitenkin mihin toiminto täytyisi kirjottaa, joten se auttoi etenemisessä.

Loppupäivän käytin sertifikaattialueen läpikäyntiin. Alue käsitteli lisää luokkien ja elementtien laajennuksia. Tämä auttoi tarkentamaan sitä, että tehtävässäni täytyi laajentaa taulun toimintaa. Sen sain menemällä taulun valintoihin, valitsemalla metodit ja laajentamalla siellä olevaa koodia. Tässä tehtävässä metodi olisi `init()`. Sertifikaattimateriaalissa käydään myös läpi sääntöjä, jotka on otettava huomioon laajennuksia tehtäessä. Näitä ovat tärkeimpinä viittaussäännöt ja se, että vain muuttujia, joissa on laajennuksen salliva ehto hyväksyttynä, voidaan laajentaa.

Torstai 28.4.2022

Jatkoin viikkotehtäväni tekemistä. Sain eilen ideaa siihen, miten ratkaisisin sen ja tänään aloin jatkamaan tehtäväni ratkaisemista. Minulla oli idea, että tehtävänä oli laajentaa `init()`-metodin toimintaa, joten aloin tekemään sitä. Ensiksi minun täytyi käyttää Query-kyselyä, jolla määrittelin haettavien kenttien tiedot. Tiedot olivat yritykset ja yritysten laitteet. Kun olin tuonut tiedot onnistuneesti käyttöön, minun täytyi testata, että saisinko käsin yhden yrityksen laitteet näkyviin. Laitoin manuaalisesti kovakoodaamalla kyselyn arvoksi esimerkkiyrityksen "DE-001" tiedot, sillä tiesin että sille oli tallennettu laitteita. Käynnistin sovelluksen ja mennessäni välilehdelle, se näytti oikean yrityksen kaikki laitteet listattuna.

Tämän jälkeen minun tulisi saada valinta käyttöön eli se, että klikkaisin yritystä ja mennessä laitteet-välilehdelle, se näyttäisi tämän yrityksen laitteet. Loin tätä varten uuden funktion, joka käsittelee `if`-lauseella sen, että onko yritys valittu. Jos on, se palauttaa kyseisen yrityksen laitteet. Jos ei, se palauttaa kaikki. Minun täytyi vielä tehdä funktio, joka palauttaisi klikatun yrityksen tiedot `if`-lauseelle. Testasin ensiksi näiden funktioiden toimintaa kovakoodatulla yrityksen nimellä ja varmistin niiden toiminnan. Kun tämä toimi, lähdin muuttamaan sitä niin, että yrityksen arvo tulisi klikkauksesta. Klikkauksen arvon sai muuttuun `element.args().record()`-lauseella. Tällä arvolla suodattamalla haku onnistui ja tehtäväni oli valmis. Testasin vielä tehtävän toimintaa kokonaisuudessa ja totesin sen toiminnan. Huomenna kävisin vielä kouluttajan kanssa ratkaisua läpi.

Perjantai 29.4.2022

Perjantai alkoi taas tiimikokouksella ja se eteni normaalisti. Erikoisuutena kerrottiin, että yksi työntekijöistä oli tehnyt raportoinnin eräästä työstä, jonka hän suoritti. Tämä työ oli erään nettikaupan alustan tekeminen ja hän oli tekemässä esitelmää työstä ja sen vaiheista. Tämän hän esittelisi

maanantaina. Muuten kokous noudatti normaalia kaavaa ja kiireisen kevään jälkeen työmäärä on hiukan helpottamassa, kuten jokainen kesä tapahtuu.

Kävin kouluttajani kanssa läpi tehtäväni statusta. Esittelin kouluttajalle ratkaisua ja koodia ja hän kommentoi ratkaisua. Ratkaisuissa ei ole minkäänlaista oikeaa tyyliä, mutta usein ratkaisuja pystyy kuitenkin tiivistämään ja selkeyttämään. Kokonaan uudessa koodikielessä on monia eri komentoja, joista en tiedä, ja koska kieli ei ole yleisesti käytetty, niitä ei ole kovin helposti löydettävissä edes netistä. Ulkopuolisen vinkit ovat siis todella tärkeitä kehittymisessä ja työssäkin tuotetun koodin arvioi aina toinen työntekijä. Tätä ratkaisua tarkastellessa kouluttajan huomio oli se, että koko toiminnallisuuden pystyisi suorittamaan yhdessä funktiossa, joka samalla tekisi toiminnasta luotettavamman ja nopeamman. Muuten tämä oli hyväksytty suoritus ja sain tehtäväksi jatkaa sertifikaattikoulutusta. Sovelluksen korjaaminen oli tehty sillä, että kopioin vain muista funktioista rivit ja siirsin ne suoraan pääfunktioon.

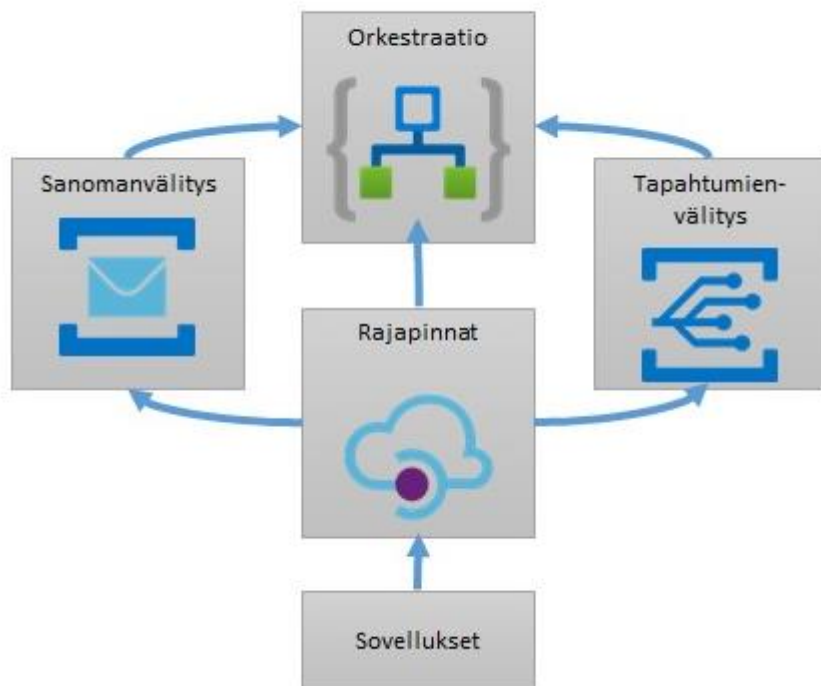
Jäin vielä omatoimisesti harjoittelemaan DevOps ympäristön toimintaa ja versiohallintaa. Kun olin korjannut kaikki koodin virheet ja saanut kaikki hälytykset pois, uusin versio oli valmis laitettavaksi versiohallintaan ja katsoa rakentuuko se siellä. Työnsin Visual Studiosta uuden version versiohallintaan ja kävin katsomassa, että siinä on kolme päällekkäisyyttä vanhan version kanssa. Muistin miten päällekkäisyydet korjataan, mutta en sitä, mistä niitä pääsi korjaamaan. Käytin yli tunnin siihen, kunnes löysin neuvoa, että minun täytyi vaihtaa haaraa ja sieltä yhdistää oma haarani päähaaraan. Siinä vaiheessa tulee päällekkäisyyksien yhdistäminen ja sain ratkaistua ongelmat. Tämän jälkeen minun työhaarani yhdistyi tuotantohaaraan ja se lähti automaattisesti kääntymään DevOpsista. Kymmenen minuutin odotuksen jälkeen kääntyminen oli onnistunut ja sain huoahata helpotuksesta. Viikon työt olivat tallessa ja toiminnassa.

Viikkoanalyysi

Tämä viikko oli muista poikkeava. Minulla oli integraatiokoulutusta, mutta muuten viikon tekemiset olivat hyvin omaehtoista tekemistä harjoitusten ja sertifikaattikoulutuksen kanssa. Integraatiokoulutus tuntui jopa hieman rikkovan omaa tekemistä ja se oli toteutukseltaan hieman hankala. Kuitenkin oli hyvä nähdä, miten integraatiot toteutetaan. Eniten kuitenkin opetti oma tekeminen ja oma tiedonhakeminen. Tehtävät, jotka minulle oli annettu, oli tehty siten, että tehtäviin löytyisi mahdollisimman vähän lähteitä ja joutuisin selvittelemään ratkaisua itse.

Azuren integraatioita on yleensä neljää erilaista. Azuressa on neljä erilaista sovellusta, josta voidaan tehdä integraatio ja näitä ovat API management, logic apps, service bus ja event grid. Jokai-

nen näistä voidaan integroida ulkopuolelle, ja jokaiselle näitä on oma työkalunsa. Näitä ovat orkestrointi, sanomanvälitys, tapahtumienvälitys ja rajapinnat. Meidän käyttämä integraatio oli rajapinnat eli REST API, joka palauttaa tiedot JSON muodossa. (Marrocos 2020).



Kuva 5. Azure integration servicen palikat (Ryhänen, J. 2022)

Logic appit ovat low-code-työkaluja, jotka kehittävät bisnesprosesseja ja datavirtoja orkestraatioina. Käyttämällä Logic Apppeja, voidaan työnkulkua laajentaa muokatuilla ominaisuuksilla. Service Bus kuljettaa sanomia ja jonoja, jossa integraatiokerros vastaa tiedon välityksestä ja koordinoimista. Event Grid toimii samalla periaatteella, mutta se vuorostaan välittää eteenpäin tapahtumia. (Devisioona, 2022). Työpaikallamme on oma integraatiotiimi, joka työskentelee näiden kanssa ja ne ovat tärkeitä itselle tietää. Tärkein meidän työmme kannalta on kuitenkin rajapinnat, sillä kuten kuvioista 7 näkyy, se on ensimmäinen piste, johon sovellukset linkittyvät. (Ryhänen 2022).

Oman sovellukseni harjoitustehtävät rajoittuivat tällä viikolla rajaukseen ja kyselyihin. Kyselyillä saa haettua tiettyä dataa listoista. Kysely on ensin alustettava ja sille on annettava tietolähde QueryDataSource, sekä arvojoukko QueryRange (Microsoft 2018b). Kun tietolähde ja arvojoukko on annettu, kyselylle tulee antaa haettava arvo. Arvon voi määrittellä monin eri tavoin, mutta tässä tehtävässä arvoksi tuli haettava aktiivisena olevan yrityksen nimi. Yrityksen nimellä pystyimme rajaamaan kyselyä ja funktio palauttaa haluamamme arvot. Tässä tapauksessa arvot olivat yrityksen nimi ja sen nimiin kirjatut laitteet. (Microsoft 2022e).

Tiimipalaverit jatkuivat joka toinen päivä. Tällä viikolla minulle ei vielä löytynyt asiakasprojektiä. Töiden määrä vaihtelee niin paljon päivittäin, että sopivaa pitkäaikaista projektia, johon lähtisin työskentelemään mentorin kanssa, ei vielä ole. Tahti alkaa myös hidastumaan kesää kohden, sillä asiakasyrityksissä työntekijät alkavat lomailemaan. En ota stressiä tilanteesta, vaikka haluaisinkin jo kovasti oikeita projekteja. Tiedän, että tekemistä kuitenkin tulee riittämään, kun vauhti taas kiihtyy. Etätöiden aikana aloittamisessa on ollut myös se haasteensa, että keneltä kysyy neuvoa. Toimistolla muilta voisi kysyä apua, mutta etänä kynnyks on ollut hieman korkeampi. Hitaampi aloitus tahti siis voi olla hyödyksi, etten jää ilman apua suuren työmäärän alle.

Tietojen hakeminen kyselyillä ja rajaaminen on tärkeä taito, sillä sitä käytetään myös esimerkiksi tietokantakyselyissä, kun tietoa halutaan lisätä, muokata tai poistaa. Viikon tehtäväni, eli vain tiettyjen tietojen näyttäminen on varsin mahdollinen työ, joka täytyy suorittaa asiakasprojekteissa useaan kohtaan. Tämä viikko opetti itsenäistä työskentelyä ja tiedonhakua, sekä versiohallintaa DevOpsiin Tein useita virheitä, jouduin jopa palauttamaan työni aiemman version versiohallinnasta, koska sain sen niin sekaisin, mutta se on opettanut paljon. Sain kuitenkin tehtävän tehtyä, tehostin sen toimintaa ja sain sen versiohallintaan. Tulevalle viikolle minulle ei ole laitettu koulutuksia, mutta maanantai lähtee käyntiin kouluttajan kanssa, joten tulen varmasti saamaan lisää tehtäviä suoritettavaksi.

3.5 Seurantaviikko 5

Maanantai 2.5.2022

Maanantain tiimikokouksessa kollega esitteli työnsä saavutuksia muille. Työn projektinimi oli ShopListFloor ja se oli sovellusalusta nettikaupalle. Hän esitteli mitä työkaluja ja tekniikoita sen käyttämiseen on tehty, mitä toiveita asiakkaalla on ollut sen suhteen, sekä miten hän on tulkinut toiveita. Samalla hän esitteli miten yhteyden ylläpito ja kehitys on mennyt asiakkaan kanssa eteenpäin. Lopputulos oli toimiva nettikaupan sovellus, josta pystyi tilaamaan tuotteita ja ostoksista kirjautui automaattisesti järjestelmään tiedot, sekä raportit ja asiakkaalle kuitit. Sovellukseen oli yhdistetty nettipankkien maksutoiminnot, sekä asiakas pystyi lisäämään ja muokkaamaan tuotteita helposti. Mobiilisovelluksen näkeminen oli todella mielenkiintoista, sillä se antoi taas yhden näkökulma siihen, mitä minun työni saattaa tulla olemaan.

Tämän jälkeen pidin kokouksen kouluttajani kanssa, jonka kanssa keskustelin edistymisestä ja hän antoi uuden tehtävän tälle viikolle. Tehtävä täytyi tehdä Custom service -integraatorajapinta, jonka kautta pystyisi luomaan uusia asiakkaan laitteita. Tiedot pitäisi pystyä lisäämään JSON-muodossa ja parametreina toimisi asiakasnumero ja laitetunnus. Tämän lisäksi pitäisi halutaan vaihtoehtoinen

parametri `TaxExemptNumber` ja virheenkäsittely virheiden tapahtuessa. Tiedot voisi lisätä esimerkiksi lähettämällä kutsun `Postmanilla`.

Aloitin tehtävän tekemisen taas hakemalla tietoa siitä, millaisia luokkia tarvitsen uuden toiminnan rakentamiseksi. Löysin muutamia ohjeita asiasta, mutta yksi vaikutti parhaalta. Sen mukana olisi tehtävä kolme luokkaa nimeltään `request`, `response` ja `service`. Tämän lisäksi vaadittaisiin `Service` objekti, sekä `Service` ryhmä. Kutsuun, vastaukseen ja palveluun tulisi koodia, joka ottaisi vastaan kutsun, tekisi vastauksen ja `service` lähettäisi vastauksen. `Service` objekti olisi se, jota kutsuttaisiin ja se lähettäisi vastauksen rajapintaan. Ryhmä kokoaa kasaan useampi samanlaisia objekteja ja tässä tapauksessa sinne liitettiin `service` objekti.

Tiistai 3.5.2022

Tänään minulla ei ollut virallisia kokouksia, vaan sain lähteä suorittamaan tehtävää. En tehnyt eilen mitään virallista, vaan vasta tiedonhakua ja suunnittelua. Ensiksi lähdin toteuttamaan `request`-luokkaa. Tässä tehtävässä tuli kutsua kolmea eri tietoa, joten alustin kolme muuttujaa nimeltään `deviceId`, `accountNum` ja `TaxExemptNumber`. Jokainen näistä muuttujista ottaa vastaan kutsun merkkijonon arvon ja vertaisi sitä olemassa olevaan luokkaan. Jos arvolla löytyy vastaus, se palauttaa sen.

`Response` luokkaan tuli kirjoittaa vaihtoehdot sille onnistuisiko kutsu, epäonnistuiko se vai tulisiko siitä debug-ilmoitus. Tähän täytyy vain kirjoittaa jokaiselle tapaukselle oma vastauksensa, joita `service`-luokka sitten kutsuu. Vastaukseen laitoin vielä palautuvana merkkijonona `JSON`-kutsusta saadun arvon.

`Service`-luokka on vastuussa siitä, että se lukee `requestin` arvot, käsittelee tietoa ja asettaa arvot `reponse`-objektiin. Kun lähtee luomaan `service`-luokkaa, siihen täytyy luoda aluksi metodi, jota kutsutaan ja joka ajaa toiminnot. Tein omani nimellä `createCustomerDevice()`, sillä harjoituksen ideana on luoda uusi asiakastietokone rajapinnan kautta. Testimielessä loin vaan koodin, joka palauttaa "Hello World"-merkkijonon, jos testi on onnistunut ja jos testi epäonnistuu, se palauttaa virheviestin. Myöhemmin tähän tulisi kirjoittaa toiminnallisuus, joka vie rajapinnasta tulleet tiedot tietokantaan.

Keskiviikko 4.5.2022

Aloitin työpäivän tiimikokouksella, jossa ei tällä kertaa ollut asiaa minulle. Sen jälkeen minulla oli tapaaminen kouluttajan kanssa ja kävimme läpi tehtäväni statusta. Teimme yhdessä debuggausta ja katsoimme, että testidata näyttää menevän läpi. Katsoimme yhdessä, kuinka integraation luonti menisi `Azureen`, sillä se oli itselleni vielä hieman epävarma aihe.

Ensiksi testisovellukseni täytyi käynnistää ja sieltä valita järjestelmäasetuksista kohta nimeltään Azure Active Directory applications. Tämä on näkymä Azuressa, jossa on kaikkien tilaajien id:t, nimet ja määrittelyt. Minun täytyi lisätä tänne uusi käyttäjä omalla Azuren id:lläni ja antaa sen oikeudeksi admin. Tämän jälkeen kyseistä id:tä käyttämällä voisi luoda kutsuja rajapintaan ja sen pitäisi teoriassa palauttaa tietoa. En ollut kuitenkaan vielä siinä vaiheessa, että toiminto olisi valmiina testattavaksi. Minun täytyi vielä lisätä Service objekti ja ryhmä ja tämän jälkeen voisin koittaa kutsujen lähettämistä.

Visual Studiosta tuli lisätä uusi Service. Se tulee nimetä asianmukaisesti ja tämän jälkeen laittaa sama nimi myös ulkoisesti näkyväksi. Tämän jälkeen palvelun alle on lisättävä uusi palvelun toiminta. Toimintaan on lisättävä se nimi, minkä olet antanut Service luokassa metodille, eli tässä tapauksessa createCustomerDevice. Tämä jälkeen on vielä luotava Service Group, jonka alle lisäsin aiemmin luoman Servicen. Tällä tavalla oikeat kutsut pysyvät järjestyksessä. Nyt koko toiminto olisi kutsuttavissa rajapinnasta.

Osallistuin iltapäivästä vielä firmani DevOps yhteisön aloituskokoukseen. Siellä oli keskustelun aiheena, kuinka yhtenäistäisimme firmamme DevOps käytäntöjä eri yksiköiden välillä ja kehittäisimme uusia käyttötapoja DevOpsiin. Kokous oli siitä mielenkiintoinen, että kuulin osaavien ihmisten keskustelua aiheesta ja mielipiteitä ja tämä auttaa itseänikin pääsemään maailmaan sisään paremmin. Olen itsenäisesti vain harjoitellut DevOpsia ja se ei vastaa taaskaan ihan kaikkienensa todellisuutta. Seuraava DevOpsin yhteisön kokous olisi kuukauden päästä. Olen iloinen, että saan aktiivisuuteni ja kiinnostukseni mukaan liittyä eri toimintoihin mukaan.

Torstai 5.5.2022

Aloitin heti aamun kouluttajani kanssa ja kävimme paremmin läpi rajapinnan kutsumista. Ennen kuin pystyisin tekemään kutsun, minun täytyisi saada valtuus Azurelta. Se onnistuisi tekemällä pyyntökutsua rajapintaan käyttämällä aiemmin asettamallani käyttäjä id:llä ja salasanalla. Määränpääksi tuli asettaa testikoneeni osoite. Tämä palautti valtuutuskoodin, jolla pystyn tekemään suoria kutsuja rajapintaan.

Minun tapauksessani haluan luoda uuden asiakastietokoneen tietokantaan ja kutsuttava osoitteeseen tulisi laittaa loppuosa `"/api/services/JahServiceGroup/JahService/createCustomerDevice"`. Jah tulee nimikirjaimistani, jota käytän luokkien yksilöimiseen. Viestin lähettämiseen asetin valtuutuskoodin ja viestin sisällöksi JSON-kutsun, joka sisälsi tarvitsemani kolme arvoa. Kutsu meni läpi ja tämä antoi vastauksena onnistuneesta kutsusta olevan merkkijonon, eli `"Hello World"`. Seuraavaksi olisi vielä luotava rivit, jotka veisivät lähetettävät arvot tietokantaan, mutta siihen en ehtinyt tänään.

Kävin loppupäivän läpi sertifikaatin opetusmateriaalia, joka käsitteli rajapintoja ja datan käsittelyä, sekä sen viemistä muiden sovellusten käyttöön, kuten Power Appseille. Microsoftin materiaali antaa monia hyviä käytännön esimerkkejä virallisista käytännöistä ja toiminnoista, joita en vielä edes tarvitse. Minun pitäisi sopia aika viralliseen aikaan sertifikaatin tekemiseen ja käydä vielä ennen sitä materiaalit kerralla kunnolla läpi.

Perjantai 6.5.2022

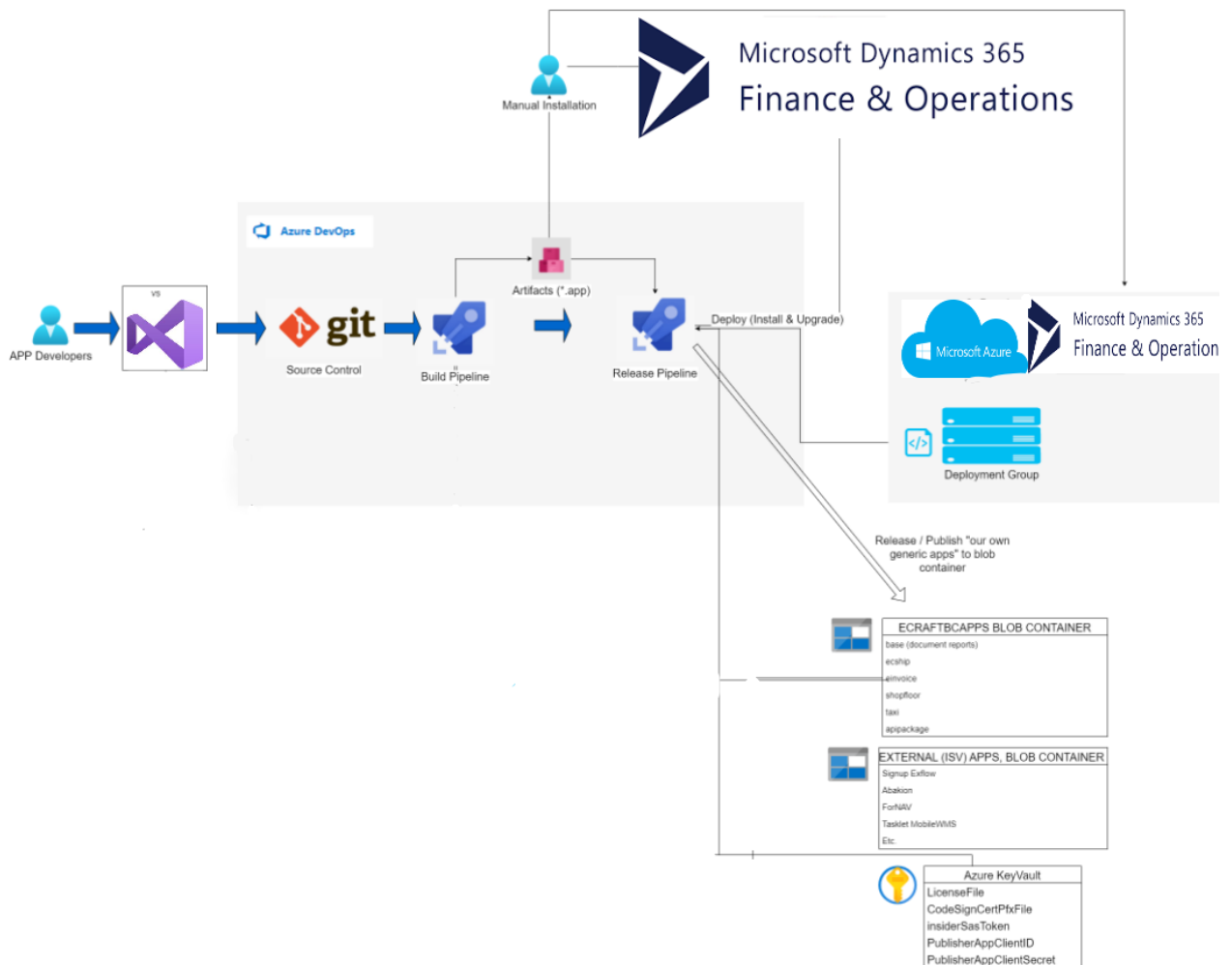
Perjantaina meillä oli pilvipalveluiden osaston virkistäytymispäivä. Näin etätyöaikana virkistäytymispäivät ovat täysin uudessa arvossa. Päivän virallinen osuus oli DevOpsin esittely eri osastojen puolesta ja muuten siihen liittyi ruokailua ja yhteistä tekemistä, jolla tutustui ihmisiin. DevOpsin tehokas käyttäminen ja jatkokehittäminen on tärkeä aihe tulevaisuuden kannalta ja siihen panostetaan tällä hetkellä paljon. Kaiken kaikkiaan päivä oli todella hyvin järjestetty ja tuli tarpeeseen senkin kannalta, että sai tutustua kaikkiin ihmisiin vihdoinkin paikan päällä oli parasta antia.

Viikkoanalyysi

Tämä viikko oli paljon enemmän itsenäistä tekemistä. Tiesin, että aihe liittyi rajapinnan tekemiseen, mutta en odottanut sen vievän näin paljoa aikaa. Siitä riippumatta, tämä viikko on ollut varmaan opettavaisin tähän mennessä. Oma tiedonhaku ja kokeileminen on aina mielenkiintoisinta. FO ympäristö alkaa tulemaan tutuksi ja tuntuu, että vihdoinkin miten ympäristö koostuu ja mitä eri lomakkeita, tauluja, luokkia ja moduuleja siellä on. Päiväkirjan pitäminen on myös suuressa osassa siinä, että omaa tekemistä täytyy reflektoida ja siitä täytyy kirjoittaa analyttisesti. Tällä hetkellä tuntuu, että suurin vaikeuteni on X++ koodikieli ja sen syntaksit. Osaan kirjoittaa yksinkertaista koodia, mutta syvempi ymmärtäminen sen kaikista ominaisuuksista on vielä vajaata.

D365FO:sta on mahdollista tehdä integraatioita kuudella eri tavalla. Näitä ovat Power Platform integraatiot, OData, Rest api, mukautettuja palveluita, ulkoisia web palveluita kutsuvan tai Excel integraatio. Integraatiot voidaan jakaa vielä synkronisiin ja asynkronisiin integraatioihin. Synkronisia kutsuja tehdessä, käyttäjä lähettää kutsun ja niin pitkään kuin kutsu on voimassa, hänen täytyy odottaa vastusta. Asynkronisissa kutsuissa kutsu tehdään ja vastaus tulee jossain vaiheessa, mutta käyttäjä pystyy jatkamaan sovelluksen käyttöä sitä odottaessa. Minun tekemäni mukautettu palvelu on asynkroninen, eli kun kutsun sitä postmanin kautta, minun täytyy odottaa vastausta ennen muuta tekemistä. Asynkronisia kutsuja käytetään useimmiten integraatioihin, joissa kulkee suuri määrä dataa ja tietoa. Hyvänä sääntönä voisi käyttää rajaa, että jos tietoa on yli muutaman sadan tuhannen, tulisi käyttää asynkronista kutsua. (Microsoft 2022f).

DevOps tarjoaa versiohallinnan ja sovelluksen kehityspotken. D365FO projektit ovat yhdistetty DevOpsiin git-versiohallinnalla, josta ne päivittyvät DevOpsin muistiin. DevOpsiin määritellään sovellusten kehityspotkia, jotka rakentavat ja määrittelevät sovelluksen toiminnan. Kehityspotkeen pysyy valitsemaan ajettavia toimintoja, jotka määritellään yaml-tiedostoon. Tämä yaml-tiedosto listaa komennot, jotka ajetaan sovelluksen rakennusvaiheessa. Näitä toimintoja voi olla esimerkiksi testien ajaminen, PowerShell-skriptin lisääminen ja artefaktien julkaiseminen. (Microsoft 2022g). Tein DevOpsin omasta käyttötapauksestani havainnollistavan kuvan (Kuva 6). Käytin sitä yhdessä koulutyön esitelmässäni, mutta se selkeyttää itselleni oman työympäristöni toimintaa. Varmistin vielä osastomme DevOps-vastaavalta, että kaavion toiminta on oikein.



Kuva 6. Azure DevOps toimintaympäristö työssäni

Yrityksen DevOps yhteisö tapasi tällä viikolla ensimmäistä kertaa ja aiheena oli se, kuinka voisimme yhtenäistää toimintaa. Selkeimpinä kehityksen paikkoina olisi se, että koodit tulisivat samalla tavalla DevOpsiin. Tällä hetkellä osa käyttää DevOpsin omaa versiohallintaa, osa BitBucketia. Kehityspotkujen toiminnan yhtenäistäminen ja käytäntöjen kirjaaminen olisi myös ensiarvoisen tärkeää. Tällä hetkellä olemme tilanteessa, jossa jokaisella osastolla on oma DevOps-osaajansa, jotka ovat itse määritelleet asetukset parhaaksi näkemällään tavalla.

Mäntyneva (2018) mainitsee kirjassaan ”Hallittu Projekti” suunnitteluvaiheesta vanhan viisauden ”hyvin suunniteltu on puoliksi tehty”. Esimerkkeinä projektin suunnitellun tärkeistä vaiheista Mäntyneva mainitsee suunnittelun organisoinnin, taustaselvityksen, rajaukset, sidosryhmien arviointi, suunnitelman arvioinnin ja viimeistelyn. Tätä seuraa projektin aloitusseminaari, tiedottaminen projektin käynnistämisestä ja sen toteutus. Käynnistyksen jälkeen projektille tulee määritellä tavoitteet ja kulkea suunnitelmallisesti niitä kohti. Nämä tavoitteet tulisi kuvata hyvin, sekä määritellä miten asetettuihin tavoitteisiin päästään. Tässä vaiheessa suunnitellaan myös kokoonpano, vastualueet ja tehtävät. (Mäntyneva 2018, 20).

Projekti DevOpsin käytön määrittelyn yhdistämisestä hoidetaan kokonaan firman sisäisesti ja se koostuu DevOps tekijöistä ja vapaaehtoisista osallistujista. Projektilla on selkeät johtajat ja jokainen heistä on esitellyt muille, kuinka he käyttävät DevOpsia. Projekti käynnistettiin pienen ryhmän sisällä, mutta tämän jälkeen siitä tiedotettiin suuresti yrityksen yhteisessä tapahtumassa ja projektin alkutilanne esiteltiin kaikille. Projekti on nyt siinä vaiheessa, että siihen haetaan kiinnostuneita ihmisiä ja keskustelu kehitysideoista on päällä. Tulevien kokousten ajankohdat ja aiheet ovat jo määriteltä. Tavoite on selkeä, mutta keskustelua sen toteutuksesta käydään vielä. Seuraavassa kokouksessa kokoonpano, vastualueet ja tehtävät tulevat olemaan määrittelylistalla. Vaikka kyseessä on vain yrityksen sisäinen ja vapaaehtoinen projekti, on se hyvin suunniteltu alusta alkaen.

Ensi viikolla minun tulisi liittyä asiakasprojektiin. Asiakasprojektissa pääsen tekemään yksinkertaisia toimenpiteitä ja näkemään, millaista tekeminen oikeasti on. Odotan sitä jo kovasti. Minun täytyy suorittaa sertifiointin opiskelualue loppuun ja varata aika sertifiointikokeen suorittamiseen. X++ koodikielen kertausta ja harjoittelu on myös ensiarvoisen tärkeää tulevalle viikolle. Tämä viikko on ollut hyvin opettavainen, ja odotan tulevalta vähintään samaa.

3.6 Seurantaviikko 6

Maanantai 9.5.2022

Viikko lähti käyntiin totuttuun tapansa tiimitapaamisella. Tämä tiimitapaaminen oli itselleni jännittävä, sillä saisin tietää mihin projektiin ja kenen kanssa minut laitettaisiin. Esimieheni oli keskustellut muiden jäsenten kanssa ja he päättivät laittaa minut asiakasprojektiin, joka on tällä hetkellä

enimmäkseen ylläpitoa ja pieniä muutoksia. Asiakkuus on alkanut kaksi vuotta sitten ja suurin työ on siis jo toimitettu. Mentoriksi sain miehen, joka on tehnyt näitä töitä jo 25 vuotta.

Jotta voisin aloittaa ympäristön alustamisen, minun täytyy saada pääsy asiakkaan ympäristöön. Esimieheni alkoi tätä selvittämään. Asiakas vastasi vielä saman päivän aikana, että minulle on annettu oikeudet, mutta sinne täytyi vielä luoda virtuaalikone, jotta pääsisin tekemään jotain. Tutkin hetken heidän ympäristöään DevOpsista ja LCS:ssä, mutta muuten voin vain odotella.

Harjoittelin loppupäivän testiympäristössä aiemman integraationi loppuunsaattamista ja sertifikaattikoetta varten. Integraatiossa minun täytyi vielä varmistaa, että tieto tulee oikein sisään kutsusta ja että tiedot tallentuisivat järjestelmään. Sain perjantaina jo kutsun toimimaan ja sieltä tuli kolme tietoa, joiden tulisi tallentua. Ensiksi minun täytyi alustaa tulevat tiedot muuttujiin. Tämä jälkeen X++:ssa on oma toimintonsa tiedon tallentamiselle. Se on ttscommit ja tämän toiminnon jälkeen listasin tietojen tallentuvan oikeaan paikkaan. Etsin järjestelmästä mihin tietojen tulisi tallentua FO-sovelluksessa. Menin oikeaan paikkaan ja hiiren oikeaa painamalla sain taulun tiedot. Tästä tiesin, mihin ohjaisin sisään tulevan tiedon.

Laitoin uusien muuttujien tiedot tulemaan tauluihin oikeaan paikkaan. Hakemani tiedon perusteella yhdistin nämä ja debuggaamalla pystyin näkemään, tuleeko oikea tieto sisään. Kaikki näytti toimivan, joten laitoin vielä insert ja ttsend-komennot. Nyt tieto, joka tuli ulkoa, meni oikein sisään ja tallentui järjestelmään. Tämä oli viimeinen harjoite kouluttajani kanssa, mutta hän oli kipeänä, joten emme pystyneet tarkistamaan onko ratkaisu oikea. Jatkoin sertifikaattiharjoitteiden tekemistä ja lukemista odottaessani jatkotietoa asiakasprojektista.

Tiistai 10.5.2022

Asiakasprojektista ei ollut tullut tietoa aamuun mennessä. Kyselin siitä eteenpäin ja kuulin, että minun täytyisi ottaa yhteyttä meillä henkilöön, joka laittaisi minulle virtuaalikoneen käynnistymään ja loputkin admin oikeudet. Otin häneen yhteyttä ja hän lupasi hoitaa asian. Ainoa mitä pystyin tekemään, oli taas odottaminen.

Jatkoin sertifikaattiaineiston lukemista. Varasin myös ajan sertifikaatin tekemiseen. Sertifikaattikoeseen tulisi varata vähintään kaksi tuntia aikaa ja mahdollisimman tyhjä huone, joten varasin toimistosta tyhjän työhuoneen samalle ajalle. Tein myös samalla testikokeen, joka mukailisi sertifikaattikoetta ja testaisi samalla, että koejärjestelmä toimii minun työkoneellani. Harjoittelukoe oli suhteellisen haastava ja kysymykset yksityiskohtaisia. Tiesin parhaiten järjestelmästä ja koodaamisesta, mutta esimerkiksi integraatio muiden sovellusten kanssa ja tietojen printtaaminen oli vaikeaa. Sain lopuksi kolmekymmentä prosenttia oikein ja vaadittava on kahdeksankymmentä. Tiedän ainakin mitä lukea.

Keskiviikko 11.5.2022

Keskiviikon tiimikokouksessa sain kuulla, että virtuaalikoneeni on valmis ja sain alkaa toimintaan. Uuden ympäristön kanssa on aina tärkein ajaa kaikki päivitykset ja asentaa tarvittavat työkalut. Minun virtuaalikoneeni oli vanhaa mallia, joten ajoin ensimmäisenä Windowsin päivitykset. Näitä löytyi läjäpäin, joten niiden lataamisessa ja asentamisessa meni tunteja, sekä kone täytyi käynnistää usein uudestaan. Tämä jälkeen asensin Visual Studioon kaikki päivitykset. Mentorini käski myös asentamaan AzCopy ja SQL Service.

Kun kaikki oli valmiina, odotin jotta mentorillani olisi aikaa ympäristön tietokannan importoitavaksi. Tämä oli aihe, josta en tiennyt ollenkaan, vaan kuvittelin tietokannan tulevan koneen asennuksen mukana. Mentorini antoi komennot, joilla tekisin tietojen siirron ja aloimme tekemään sitä. Ensiksi DevOpsista saisin myös haettua ja asennettua koko ympäristön, mutta siinä saattaisi mennä useita tunteja tietokannan laajuuden vuoksi. Siksi käytimme AzCopy-työkalua. Hain linkin LCS:stä ja ajamalla AzCopy komennon PowerShellissa, tietokanta tuli muutamassa minuutissa Backpac-muodossa. Tämän jälkeen se olisi purettava komentokehityksestä hyväksikäyttämällä SQL Serviceä. Paketin purkamisen jäi koneelle pyörimään ja jätin sen yöksi päälle, sillä työaikani ei riittänyt sen odottamiseen.

Torstai 12.5.2022

Otin aamulla yhteyden virtuaalikoneelle tarkastaakseni missä vaiheessa asennus meni, ja näemmä se purki pakettia edelleen. Asioita oli kuitenkin mennyt yön aikana eteenpäin, joten en oletanut sen olevan jumissa. Jatkoimme mentorini kanssa työtilanteesta juttelemisella ja hän näytti jo ensimmäisiä työtehtäviä, joita mahdollisesti pääsisin tekemään ensimmäisenä. Siellä oli muutamia pieniä töitä, mutta kaikki koodimuutokset tuntuvat varsinkin alkuun pelottavalta, eikä työohjeita ole näissä tehtävissä kovinkaan paljon.

Odottaessani kertosin taas sertifikaattikokeen materiaalia. Tällä kertaa keskityin enemmän integraatioihin ja muihin osa-alueisiin, joissa pärjäsin testikokeessa huonosti. Näiden osa-alueiden opettelu on vaikeinta, sillä en pääse harjoittelemaan näitä edes testiympäristössä. Tiedot on vain periaatteessa muistettava ulkoa, jotta kysymyksistä voi selvittää.

Iltapäivästä paketin asentaminen oli mennyt kokonaan läpi. Aukaisin Microsoftin SQL Serverin, sillä minun täytyisi siellä vielä alustaa uusi tietokanta. Lausekkeet olivat minulla valmiina, mutta ajoin niitä vain vähän kerrallaan, saadakseni kiinni mahdolliset poikkeukset. Kaikkien taulujen luonti onnistui hyvin ja uusien tietojen linkitys onnistui. Minulle tuli ainoastaan ongelmia, kun uusi tietokanta piti laittaa vanhan tilalle. Ihmettelimme tätä yhdessä mentorini kanssa ja hänkään ei osannut heti sanoa mistä ongelma tulee. Lopulta se oli niinkin pientä, että aiemmat ikkunat SQL

Serveristä täytyi sulkea, sillä se antoi virheen. Kun suljin avoimet välilehdet, pystyin tuomaan uuden tietokannan vanhan tilalle. Nyt kaiken tulisi olla valmista työn aloittamista varten.

Perjantai 13.5.2022

Perjantai alkoi tiimikokouksella, jossa keskustelimme nykyisestä työtilanteesta. Kävimme mentorini ja esimieheni kanssa sitä, että mitä lähtisin tekemään. Löysimme todella yksinkertaisen työn, jossa pitäisi vaihtaa vain muutaman etiketin nimi. Sovimme, että tästä lähdetään liikkeelle ja saisin niin paljon ohjausta kuin tarpeen.

Kokouksen jälkeen lähdin tekemään muutosta. Tietin sain Jira-palvelusta ja siellä selostettiin, mikä työ on. Tietissä luki, että kahden etiketin nimi pitäisi vaihtaa eri muotoon. Käynnistin uuden projektin ja nimesin sen tietin mukana tulevan tunnuksen mukaan. Tämän jälkeen minun tulisi löytää sovelluksesta paikka, jossa kyseiset etiketit ovat. Aukaisin AOT-objektitaulun, josta pystyn etsimään koko sovelluksen tietoja. Lähdin etsimään taulua, josta minulla oli jo etukäteen tietoa, että etiketit saattaisivat olla siellä. Pääsin melko lähelle oikeaa ja klikkailemalla valintoja, sain oikean näkymän eteeni. Tästä pystyin tarkistamaan etiketin virallisen nimen.

Kun tiesin oikean nimen, lähdin etsimään etikettejä etikettilistalta. Sieltä näin, että kyseessä on yleisesti käytettyjä etikettejä ja jos lähtisin muuttamaan niiden nimeä, muuttaisin kaikkien vastavien nimiä. Minun siis täytyi luoda molemmille omat etiketit. Loin molemmille uudet etiketit ja laitoin niiden arvoon pyydetyn nimen. Tämän jälkeen kävin vaihtamassa oikeaan tauluun tietojen kohdalle uudet tiedot. Työnkuvassa ei kuitenkaan kerrottu, mikä olisi etiketin suomenkielinen nimi. Kysyin tätä meidän sovellusarkkitehdiltamme, joka on antanut työn minulle. Hän kysyi tätä asiakkaalta, mutta hekään eivät tienneet. Ennen kuin voisin lisätä muutokset versiohallintaan, minun täytyisi saada vastaus tähän. Jäin siis odottelemaan viikonlopun yli.

Viikkoanalyysi

Tällä viikolla pääsin vihdoinkin asiakasprojektiin kiinni. Minulla ei ollut mitään odotuksia siitä mitä tulisi tekemään tai kenen kanssa ja se tuntui hyvin hermostuttavalta. Viikon aikana minut sijoitettiin kuitenkin vanhan asiakkaan ylläpitotehtäviin ja uskon se liittyvän siihen, kuka tuli minun mentoriksi. Mentorini on hyvin avulias, sekä opettaa minulle paljon uusia asioita, mutta myös laajentaa vanhan tietoa. Ylipäättänsä nyt tuntuu, että tiedän mitä tulisi tekemään ja minulla on tukea siihen, mutta se ei tietenkään lopeta uuden työn pelotusta.

Uutena asiana minulle tuli uuden työympäristön tuonti virtuaalikoneelleni. Tähän käytettävät työkalut olivat AzCopy ja SQL Service. AzCopy on komentokehoitteen työkalu, jonka avulla voi kopioida

paketteja tai tiedostoja, joka sisään tai ulos tallennustilasta. Minun tapauksessani käytin sitä kopioidakseni dataa asiakkaan Azure DevOpsista. AzCopy täytyy ladata koneelle zip-tiedostona, joka sisältää suoritettavan exe-tiedoston. Tämän jälkeen sinun täytyy tietää DevOpsin SAS-token, ja pystyt ajamaan käskyn komentokehoitteelta. Tämä jälkeen sinulla on koko tietokanta koneellasi usein alle minuutissa. Koko lataamisen voisi hoitaa myös Azuren palvelusta, mutta silloin paketin lataaminen saattaisi venyä monituntiseksi. (Microsoft 2022h).

AzCopy tuo tiedoston, joka on Backpac-muodossa. Backpac on Azuren oma tiedosto, joka sisältää koko valitun palvelun metadatan ja datan. Backpac paketin purku tapahtui SQLPackagen avulla myös komentokehoitteelta ja sen käyttö oli hyvin samanlaista kuin AzCopyn käyttö. Tämä jälkeen tietokanta täytyy alustaa SQL Serverillä. (Microsoft 2022i). SQL Serverin käyttöä avasin paremmin päiväkohtaisessa käsittelyssä.

Olen käynyt opinnoissani käyttänyt usein komentokehoitetta ja se on tuttua niin Linuxin kuin Windowsin puolelta, mutta erikoisena huomiona ymmärsin, että en ole käyttänyt sitä juurikaan ohjelmien ajamiseen. Olen ehkä suoraan ohjeista kopioinut ja ajanut, mutta nyt kun sain vastaani suoritettavia exe-tiedostoja, joihin tuli linkittää haluttuja asioita, olin aivan hukassa, miten se toimiikaan käytännössä. Tein tietokannan haun ja asentamisen manuaalisesti, vaikka sen olisi voinut tehdä automaattisesti. Sen lisäksi, että siinä säästi kymmeniä tunteja, se auttoi taas ymmärtämään prosesseja ja toimintaa niin sanotulla ruohonjuuren tasolla. Eli juuri sillä tasolla, johon haluan pureutua.

Viikon suurinta antia oli myös henkilökohtaisen mentorin saanti. Tunnistan itsestäni piirteitä, että en halua tai uskalla kysyä apua ja haluan selviytyä yksin tehtävistäni. Tässä työssä se ei ole mahdollista, vaan kaikki sosiaalisuus on plussaa. Mentoroinnin hyödyt ovat myös työpaikalle selkeästi nähtävissä. Näitä ovat uuden työntekijän urakehittymisen varmistaminen, kattavan työyhteisön luominen, yhteisöoppimisen tilan luominen ja mentorointi takaa usein paremman pysyvyyden uusille työntekijöille. (Burak 2021).

Olen tavannut uusia työkavereitani firman tapahtumissa, mutta se on kuitenkin erilaista kuin virallinen työskentely yhdessä. Mentorin kautta olen saanut itselleni yhden kiintopisteen, ja hän on auttanut kertomalla muita henkilöitä, joihin minun tulisi olla yhteyksissä missäkin tapauksessa. Mentorin saanti ainakin itselleni varmistaa paremman urakehittymisen ja se tuo itselleni välittämisen tunnetta.

3.7 Seurantaviikko 7

Maanantai 16.5.2022

Maanantain tiimikokous oli peruttu esimiehen kiireiden takia, mutta sain aamusta vastauksen asiakkaalta kysymykseeni siitä, mikä tulisi muutokseni suomenkieliseksi nimeksi. Käynnistin virtuaalikoneeni, tein muutoksen ja kävin tarkistamassa, onko muutos oikein englannin- ja suomenkielisillä versioilla. Molemmilla versioilla muutos näytti oikealta, joten oli aika viedä muutokset versiohallintaan.

Tein muutoksille oman haaran tiketin numeron perusteella. Tarkistin, että kaikki tulevat muutokset ovat oikeat ja alustin ne lähetettäväksi. Tämän jälkeen muutokset pystyi viedä versiohallintaan ja tein niistä katselmointipyyntöni työkaverille. Työkaveri hyväksyi muutokset päivän aikana, joten pystyin DevOpsista vahvistamaan muutokset valmiiksi. Kun tein hyväksynnän, muutokseni lähti suoraan pääharaan ja sieltä se rakentui uuteen versioon. Seurasin sovelluksen rakentumista jännittyneenä ja vartin päästä sain ilmoituksen, että kaikki on kunnossa. Tämän jälkeen palautin tiketin sovellusarkkitehdille ja hän voisi käydä varmistamassa muutoksen.

Aloimme mentorini kanssa katsomaan seuraavia työtehtäviä. Hän löysi sopivan työn, joka oli enemmän koodausta. Tehtävä tuntui haastavalta ja pelottavalta, mutta hyväksyn kaikki työt tehtäväksi. Laitoin arkkitehdille viestiä, että voin ottaa työn vastaan ja että tarvitsen kuitenkin lisäohjeita siitä, mistä kyseinen muutos löytyy ja tarkempia speksejä muutoksesta. Hän sanoi tekevänsä sen huomiseksi.

Loppupäivän käytin taas sertifikaattimateriaaliin ja kävin läpi X++ koodausta tarkemmin, sillä tiesin sen olevan tulevan työtehtäväni aihe. Minulla ei kuitenkaan ollut tarkempaa tietoa mikä olisi tarkalleen hyödyksi, joten kävin koko osion läpi. Vähintään muistaisin myöhemmin, mistä hakea apua.

Tiistai 17.5.2022

Aamuun mennessä arkkitehti sai paremmat ohjeet työtehtävään. Hän oli laittanut kuvia siitä, mistä tarvittava muutos löytyisi. Pääsin ohjeiden avulla käsiksi paikkaan ja tarkistin, että miten se toimii nykyään. Tämän avulla pystyin tarkastelemaan mukana tulevia tietoja ja miettimään miten saisin muutoksen tehtyä. Mentorini neuvoi, että minun kannattaisi koodin puolelta alkaa debuggaamaan ja selvittää, mihin muutos koodissa tulisi.

Koulutuksen aikana minulle oli opetettu debuggausta ja sitä, miten pysähtymispisteitä laitetaan ja tietoja tarkastellaan. Etsin koodista oikean lomakkeen ja tarkastelin mitä metodeja se käyttää. Näitä metodeja läpikäydessä yritin etsiä mistä oikea tieto tulee. Tässä tapauksessa tieto oli RawMaterialDate. Etsin kyseistä tietoa ja etsin missä kaikkialla sitä käytetään saadakseni selville, mistä se saa tietonsa. Debuggauksessa oli kuitenkin ongelma, sillä järjestelmäni kaatui jatkuvasti. Mentori epäili sen olevan vain tietokoneen hitautta, joten jatkoin testaamista. Aikani ihmeteltyä pyysin häneltä apua, jotta hän voisi katsoa miten ongelma selviäisi.

Debuggauksen ongelma selvisi heti. Hän pyysi minua käynnistämään debuggauksen ja olin heti menossa väärään paikkaan. Koulutuksessa opetettiin, että sovellus tulisi käynnistää joka kerta erikseen ja tätä kautta toimintaa voisi testata. Mentorini sanoi ensimmäisenä, että kiinnitä debugkaus käynnissä olevaan sovellukseen. Kehittäjäympäristössä sovellus on taustalla jatkuvasti päällä, joka oli minulle uutta. Pystyin siis liittämään käynnissä olevan Visual Studion sovellukseen. Kun käynnistin debuggauksen, Visual Studio yhdistyi sovellukseen. Kun käytin sovellusta ja tein haluttavan toimenpiteen, sovelluksen toiminta pysähtyi laittamaani pysäytyskohtaan koodissa ja pystyin tarkastelemaan mitä tietoja siinä on mukana. Tein tätä useaan kohtaan ja sain lopulta tiedon, missä kohtaa tieto päivänmäärästä päivittyi sovellukseen.

Nyt kun minulla oli paikka suunnilleen tiedossa, pystyin tarkemmilla testeillä paikantamaan sen. Oikea paikka löytyi saman taulun yhdestä metodista. Se toi päivänmäärä sovellukseen if-else toiminnolla ja minun pitäisi laajentaa tätä toimintaa.

Keskiviikko 18.5.2022

Nyt kun tiesin paikan, pystyin laajentamaan sen toimintaa. Olin saanut myös arkkitehdiltä lisää ehtoja muutoksen tekemiseen. Tarkennukset liittyivät käyttäjän valitsemiin ehtoihin. Kun lähdin tekemään metodille muutosta, minun täytyi aluksi tehdä siitä laajennus. Tämän jälkeen voisin jatkaa sen toimintaa uusilla ehdoilla. En kuitenkaan löytänyt mistään neuvoa ja monen yrityksen jälkeen päädyin kysymään apua mentorilta.

Metodien laajennuksessa pitää käyttää komentoa, joka tuo alkuperäiset tiedot mukana. Muissa koodikielissä olen tottunut, että se on käsky `super()`, mutta `X++:`ssa on sen lisäksi oma käskynsä nimeltään `next`. Minun täytyi `nextillä` alustaa metodi uudestaan ja antaa sille parametrit. Tämän jälkeen loin if-else ehdot, jolla koodi toimii ja aloin hakemaan tarvittavia tietoja. Ehdin taas hetken miettimään, etten ole varma, miten tietojen hakeminen menee, kun mentorini kertoi, että se tehdään `selectillä` SQL komentojen mukaisesti. Käyttämässäni koodikielissä olen käyttänyt hyvin harvoin SQL komentoja, joten tämäkin oli uutta. Käytännössä se on kuitenkin samaa, jota olen harjoitellut esimerkiksi koulussa tietoverkkojen opintojakson aikana.

Kun olin löytänyt näkymän, joka minun piti palauttaa, aloin hakemaan siihen tietoja. Visual studio onneksi ehdottaa löytyviä tietoja ja se helpotti tekemistä. Katsoin vaatimuksista, että tieto vaaditaan `ProdId:n`, `OpriId:n` ja `OpriNum:in` perusteella, joten alustin nämä. Näiden tietojen perusteella sain oikean päivänmäärän ja pystyin antamaan sen vastauksena alkuperäiselle metodille eli `RawMaterialDate:lle` ja `Time:lle`. Testailin toimintaa vielä sovelluksen puolelta ja se tuntui hakevan päivänmäärän tiedon oikeasta paikasta ja asetti se oikein. Lähetin arkkitehdille viestin, että muutos on valmis testattavaksi. Alla on kuva (Kuva 7), joka havainnollistaa laajennuksen koko toiminnan.

```

1  [ExtensionOf(classStr(ProdUpdCostEstimation))]
2  final class ProdUpdCostEstimation_EcrRaute_Extension
3  {
4
5      1 reference
6      protected void setBOMRawMaterialDate(
7          ProdTable      _prodTable,
8          ProdBOM         _prodBOM,
9          ProdRoute       _prodRoute)
10     {
11         next setBOMRawMaterialDate(_prodTable, _prodBOM, _prodRoute);
12
13         if (_prodBOM.OprNum && _prodRoute.FromDate)
14         {
15             if(_prodBOM.ProdLineType == BOMType::Vendor && _prodBOM.EndSchedConsump==0)
16             {
17                 ProdRouteSchedulingView prodRouteSchedulingView;
18                 select firstly prodRouteSchedulingView
19                     where prodRouteSchedulingView.ProdId == _prodBOM.ProdId &&
20                        prodRouteSchedulingView.OprId == _prodRoute.OprId &&
21                        prodRouteSchedulingView.OprNum == _prodBOM.OprNum;
22
23                 _prodBOM.RawMaterialDate = prodRouteSchedulingView.FromDate;
24                 _prodBOM.RawMaterialTime = prodRouteSchedulingView.FromTime;
25             }
26             else if(_prodBOM.ProdLineType == BOMType::Vendor && _prodBOM.EndSchedConsump==1)
27             {
28                 ProdRouteSchedulingView prodRouteSchedulingView;
29                 select firstly prodRouteSchedulingView
30                     where prodRouteSchedulingView.ProdId == _prodBOM.ProdId &&
31                        prodRouteSchedulingView.OprId == _prodRoute.OprId &&
32                        prodRouteSchedulingView.OprNum == _prodBOM.OprNum;
33
34                 _prodBOM.RawMaterialDate = prodRouteSchedulingView.ToDate;
35                 _prodBOM.RawMaterialTime = prodRouteSchedulingView.ToTime;
36             }
37         }
38     }
39 }

```

Kuva 7. Koodi, jolla laajensin lomakkeen toimintaa

Ensimmäiseen koodimuutokseen lähteminen pelotti hirveästi. Huomasin kuitenkin, että melkein vaikeampi tällä kertaa oli paikantaa muutoksen lähde. Käytin siihen itsessään päivän ja muutoksen tekeminen meni alle päivässä. FO-järjestelmä kätkee sisäänsä niin valtavan määrän tietoa, että kaiken sen hakeminen on välillä kuin labyrintissa kulkisi. X++ koodikielenä yhdistää monien kielten ominaisuuksia ja niiden ominaisuuksien tietäminen on ainakin itselleni vielä vaikeaa. Uskon kaiken selkenevän työn tekemisen aikana.

Torstai 19.5.2022

Kävin vielä tekemääni koodimuutosta läpi opetellakseni sen toimintaa paremmin. Välillä työt tulee vain tehtyä ja asiat toimivat, mutta lopuksi ei ole itsekkään varma asiasta. Nyt kävin askel askeleelta debuggerilla läpi, miten tieto tulee koodiini ja miten se käsittelee ja päivittää sen askel askeleelta. Tämän muutoksen tekeminen ja ymmärtäminen on iso askel työni kannalta ja se antaa kaavan siihen, miten tulevaisuudessa lähestyn samanlaisia muutoksia. Koodi on selkeä ja nyt osaan selittää askel askeleelta, miten koodi etenee.

Odottelin edelleen, että arkkitehti käy testaamassa muutoksen ja antaisi palautetta siitä. Uusia töitä ei tässä asiakasprojektissa ole, joten koen parhaaksi jatkaa itseopiskelua. Sertifikaattikoe on jo viikon päästä, ja vaikka kukaan ei oleta, että siitä pääsisi ensimmäisellä kerralla läpi, haluan antaa parhaan panokseni. Harjoittelin tällä kertaa uudestaan laajennusten tekemistä. Kyseisessä materiaalissa on esimerkiksi nextin käyttö mainittu, mutta vain muutamalla lauseella. Tämän takia se on mennyt itseltäni täysin ohi. Kävin koko kappaleen ajatuksella läpi löytääkseni muutkin tällaiset asiat, joita tulevaisuudessa saattaa tulla vastaan.

Mentorini kanssa oli myös keskustelua muuttujien ja varsinkin enumien käytöstä. Enumeja löytyy monesta kielestä ja meille se opetettiin nopeasti vain yhtenä muuttujana. Enumi toimii kuitenkin enemmän luokkana. Kun enumin alustaa, sen alle voi listata erilaisia arvoja, kuten esimerkiksi värejä. Tämän jälkeen kyseisen enumin kautta voi kutsua muuttujaan väriä esimerkiksi "Color.Black"-viittauksella. Mentorini opetti, että hän ei ikinä hyväksy kovakoodattua koodia, joten esimerkiksi enumien käyttö on hyväksyttävä tyyli.

Perjantai 20.5.2022

Perjantai alkoi taas tiimipalaverilla. Oma aiheeni oli tällä kertaa se, että minulla ei ole asiakasprojektia tälle hetkelle. Nykyinen asiakas ei tarvitse vielä tällä hetkellä muutoksia ja muihin projekteihin minua ei ole vielä liitetty. En ole myöskään saanut palautetta koodin testauksesta, joten en tiedä täytyykö sitä muuttaa. Esimieheni alkaa etsimään minulle sopivaa projektia tai uusia tehtäviä nykyisessä.

Päivästä minulla oli "2kk kuulumiset"- keskustelu rekrytoijan kanssa. Keskustelu tuli mielestäni todella hyvään aikaan, jotta pääsin kertomaan ajatuksiani ja jakamaan hieman stressiäni. Vaikka minua on koulutettu usean viikon ajan tehtävään, oma tuntemukseni on se, että kaikki on silti aivan uutta ja jännittävää. Mentorini on todella hyvä tuki ja opettavainen henkilö, mutta päässäni jyllää niin suuri jännityksen ja uuden oppimisen sekainen tunne. Minun on täytynyt kyseenalaistaa kaikki aiemmin oppimani, jopa koulusta saadut käytännöt ja miettiä kaikki uusiksi. Mentorini esittää hyvin yksinkertaisenkin kysymyksen, mutta minun on aina täytynyt miettiä, että tiedänkö tätä asiaa oikeasti, vai onko siinä vielä jotain uutta taustalla. Olen oppinut jo vastaamaan, että en tiedä, jolloin

mentori on selittänyt asian oman näkemyksensä ja käytettävän järjestelmän mukaan, jolloin kaikki toimintatapani ovat muuttuneet kerralla.

Kävimme keskustelussa myös läpi sitä, että nyt alkuvaiheessa olemme tilanteessa, jossa 90 % on epämukavuusalueetta ja loput 10 % tuttua. Tekemisen kautta luku muuttuu neutraalimmaksi ja työnantaja antaa kaiken tukensa ja ymmärryksen siihen, että aluksi on vaikeaa. FO ja X++ ovat sellaisia aiheita, joita ei missään opeteta, joten on aivan absurdia, että niitä sisäistäisi kukaan parissa viikossa. Luon itse itselleni aina todella kovat tavoitteet ja jos en saa ulkopuolista kommenttia, ne ajatukset alkavat paisumaan. Tämä keskustelu tuli siis todella hyvää paikkaan ja esitin vielä toiveen, että pääsisin esimieheni kanssa kahdenkeskeiseen keskusteluun. Käytännön asioita keskustelemaan, mutta myös jakamaan tuntemuksia.

Viikkoanalyysi

Tällä viikolla pääsin varsinaisen koodaamisen makuun. Vaikka minulla oli vaikeuksia X++:n omien piirteiden kanssa, tuntuu että suurin kehitys on tullut kuitenkin koodaamistavan muutoksista ja kehittamisestä. Olen aiemmin koodannut isojakin muutoksia kerralla ja toivonut, että ne toimivat oikein. Kun jokin kohta ei ole toiminutkaan, minulla on ollut ongelma, että mistä lähdän edes selvittämään. Olen kuullut monien tekevän koodia testien pohjalta, mutta mentorini opetti hänen tapansa koodata debuggaamalla.

Debuggaamista voi käyttää ongelmien paikantamiseen, ratkaisemiseen ja analysoimiseen. Debuggerilla voit ajaa koodia hidastettuna kohta kohdalta ja tarkastella miten koodi etenee. Sen avulla näet myös mistä tietoa tulee ja mihin muuttujiin tieto asettuu. Yleensä debuggeria käytetään juuri ongelmien ratkaisuun, mutta sitä voi myös käyttää rakentaessa uutta koodia. Debuggaamalla säästää aikaa koodaamisessa ja se auttaa välttämään monimutkaista koodia. Sen lisäksi virheet tulevat heti huomioon, eikä niitä kasaannu pikkuhiljaa pohjalle. Tämän lisäksi debuggaamisella saa tarkasteltua sovelluksen tietorakennetta helpoiten. Kun tarkastelet tiettyä kohtaa ja näet missä vaiheessa ja mistä tieto tulee, saat paljon paremman kokonaiskuvan toiminnasta. (Tawde 2022).

D365FO koodaus on erilaista muuhun koodaamiseen verrattuna, sillä olet tekemisessä suuren tietorakenteen kanssa. Tämän takia totesin itsellenikin olevan parhaaksi, että opettelen koodaamaan debuggaamalla. Tietorakenne on niin laaja, että sitä ei voi muistaa edes mentorini ulkoa, joten ainoa mahdollisuus ongelmien paikantamiseen ja tiedon saamiseen on debuggeri. Kirjoittaessa uutta toimintoa, pystyin katsomaan sitä tehdessä askel askeleelta, tuleeko oikea tieto sisään. Jos tietoa ei tullut, tiesin että minun täytyisi selvittää se ongelma ennen kuin pystyin jatkamaan. Tällä tavalla

muutoksen koodaus oli varsin suoraviivaista, eikö minun tarvinnut testata montaa riviä koodia kerralla ja ihmetellä jälkeenkäpäin, miksi se ei toimi. Tämän lähestymistavan opettaminen on selkeästi viikon paras asia.

Toisena selkeyttävänä asiana opin, että minun tulisi välttää kovakoodausta kaikissa tapauksissa. En ole ajatellut asiaa, sillä olen rakentanut vain pieniä sovelluksia tähän mennessä, eikä minun ole tarvinnut miettiä sovelluksen eri ympäristöjä. Kovakoodauksen ensimmäinen ongelma on juuri ympäristöjen muuttuminen. Kun sovellus viedään tuotantoon, se käy läpi monta eri ympäristöä, jossa se rakennetaan aina uudestaan. Joka kerta kun ympäristö rakentuu, muuttujien id-arvot annetaan uusiksi. Eli jo seuraavassa ympäristössä kovakoodattu arvo saattaa viitata täysin eri arvoon. Toisena on se, jos tahdot muuttaa esimerkiksi tiettyä tekstiä, joka esiintyy monessa paikassa, sinun täytyy etsiä jokainen kovakoodattu kohta muuttaaksesi sitä kaikkialla. Myös jos selvität ongelmaa ja siellä tulee vastaan kovakoodattu id, et pysty helposti selvittämään mikä komponentti on kyseessä, vaan se vaatii id:n selvittämistä. (Lee 2021).

Kovakoodaus on ollut helppo vaihtoehto pieniä sovelluksia rakentaessa, enkä ole niinkään miettinyt sen tuomia ongelmia. Tämän takia FO-järjestelmässä käytetään aina etikettejä, olioita tai enumereja arvojen tuomiseksi. Jos kaikki olisi kovakoodattu, se vaatisi ensinnäkin paljon töitä, mutta koodin lukeminen ja ongelmien selvittäminen olisi mahdotonta. Olen iloinen, että mentorini huomautti tästä asiasta välittömästi, sillä opetukseni aikana sitä ei ole kertaakaan otettu puheeksi. Opin heti alusta oikeat käytännöt ja lukiessani aiheesta ymmärrän välittömästi miksi kovakoodaaminen olisi ongelmallista.

Kaiken kaikkiaan, vaikka olen ollut stressaantunut ja väsynyt, tämä ja viime viikko ovat olleet aivan loistavia. Huomaan, että minun koulutuksestani työpaikalla on ollut apua kokonaisuuksien käsittämässä ja vaikka olen ollut hieman pyörällä päästäni, kaikesta siitä on ollut suunnattomasti tukea. Minulla on ollut kova kiireen tuntu, sillä mentorini on kysellyt edistymisen perään puolen tunnin välein, mutta ymmärrän, että sekin on ollut sitä, että hän on huolehtinut minusta. Kun olen vastannut rehellisesti, että nyt en tiedä, hän on heti auttanut ja selostanut asioita. Ohjeiden saaminen ja seuraaminen on ollut myös stressaavaa, mutta nyt minulla on todella paljon enemmän työkaluja pakisani kuin kaksi viikkoa sitten. Nyt minusta tuntuu, että tehtävä kuin tehtävä, tiedän edes mistä lähteä liikkeelle.

Tuleva viikko on viimeinen seurantaviikkoni. Minulla on myös huojentunut olo päiväkirjan tekemisestä, sillä nyt tähän tulee selkeä oppimisen kaari. Pelkäsin, että tekemiseni jäisi roikkumaan tyhjän päälle, mutta nyt kaikki on mennyt täysin suunnitelmien mukaisesti. Olen palannut päiväkirjan aiheisiin usein ja saanut sieltä tukea. Tälläkin viikolla päiväkirjan kirjoittaminen on auttanut käsittelemään ja ymmärtämään paremmin aiheita, joita olen tehnyt. Vaikka yritys on sanonut olettavansa

minun olevan itsenäinen vasta vuoden päästä ja olen tehnyt varsinaista työtä kaksi viikkoa, haluan saada mahdollisimman vahvan pohjan, jolle voin luoda työuraani.

3.8 Seurantaviikko 8

Maanantai 23.5.2022

Viikko alkoi taas tiimikokouksella. Kesään mentäessä tahti uusien projektien kanssa on rauhoittunut niin paljon, että minulle ei riitä töitä tällä hetkellä. Yhden asiakkaan projektiin voisin liittyä, mutta tunnusten saaminen sinne kestäisi noin kaksi viikkoa. Keskustelimme siis, että minulle etsitään uusia töitä ennen sitä. Tällä viikolla minulla on kuitenkin sertifiointikoe tulossa, joten sain aikaa sen harjoitteluun. Varasin myös keskiviikoksi konttorilta neuvotteluhuoneen, jonne voisin mennä rauhassa suorittamaan koetta. Kokeeseen täytyy varata huone, jossa ei ole mitään ärsykejä tai ylimääräisiä asioita näkyvillä. Pienet neuvotteluhuoneet ovat siis täydellisiä tähän tarkoitukseen ja tyhjiä kaikesta.

Aloin harjoittelemaan taas sertifiointialueen materiaalia, mutta sen lisäksi varmistin, että koneelani toimii koesovellus ja tein harjoittelukokeen. Kun varmistin, että järjestelmä on kunnossa, lähdin suoraa tekemään harjoittelukoetta. Harjoituskoe koostuu kuudestakymmenestä kysymyksestä, joista suurin osa on monivalintakysymystä. Osassa täytyy laittaa prosesseja oikeaan järjestykseen. Kokeeseen on varattu kaksi tuntia aikaa, joten kiire siinä ei ainakaan tullut. Koe käsittää kuutta eri osa-aluetta, jotka ulottuvat koodaamisesta, integraatioihin ja FO-järjestelmän rakenteeseen.

Lähdin tekemään tenttiä ja huomasin heti, että kysymykset on tehty melko yksityiskohtaisiksi ja vaikeiksi. Helpoimmat kysymykset olivat esimerkiksi siitä, miten luot uuden projektin ja koodikohtaiset kysymykset tuntuivat aika helpolta, mutta vaikeimmat olivat hyvin yksityiskohtaisia siitä, mitä luokkaa käyttäisi integraatiossa. Huomasin myös, että salausluokituksiin liittyvät kysymykset olivat hyvin yksityiskohtaisia ja niin sanottuja kompakysymyksiä. Tentin tekemiseen meni puolitoista tuntia, sillä otin aikani ja halusin lukea kysymykset tarkkaan. Sain lopputuloksena oikein 30 % kysymyksistä, kun virallinen sertifiointin raja on 80 %.

Tajusin vasta tämän jälkeen, että harjoitustentistä voi tehdä myös version, joka näyttää heti oikeat vastaukset oman vastauksen jälkeen. Tein tämän vielä ajatuksen kanssa ja huomasin, että melkein kaikki kysymykset ovat samanlaisia, mutta kaikki erilaisilla yksityiskohdilla. Tein tentin ajatuksella ja luin jokaisen kysymyksen jälkeen oikeat vastaukset ja niiden teorian. Tästä tentistä sain 45 % oikein, joka on jo lähempänä, mutta silti kaukana vaaditusta. Oikeiden vastausten lukeminen ajatuksella kuitenkin auttoi tähän mennessä eniten kehittämisessä ja näkemään pienet piilotetut asiat. Luin vielä ajatuksen kanssa teoria-aluetta, ja suoritan huomenna harjoitustentin uudestaan.

Kysymysten asiat ovat niin yksityiskohtaisia, että vastaus pitäisi todellakin tietää ja se on vielä tällä työkokemuksella vaikeaa.

Tiistai 24.5.2022

Tänäänkään minulle ei ollut vielä tullut uusia töitä, eikä aiempaa työtäni ollut ehditty testaamaan. Minulla oli yksi kokous päivästä, jossa kollegani esitteli koodimuutosta, jonka hän oli toteuttanut. Hän oli kevään aikana tehnyt suuren integraatiotoiminnon ja hän jäisi nyt kesälomalle. Jos lähiaikoina integraation käyttämiseen tulisi ongelmia, minä olisin yksi niistä, joka on nähnyt koodin ja osaisin tämän jälkeen tehdä muutoksia siihen. Tai tämän jälkeen ainakin tietäisin mistä ne löytyvät ja pystyn lähtemään selvittämään asiaa. Integraation toiminnan näkeminen oli mielenkiintoista ja syvensi omaa osaamistani siltä tasolta, että kuulin millaista tällaisen ison muutoksen tekeminen oli ollut ja näin sen toiminnan.

Tämän jälkeen aloin harjoittelemaan lisää sertifikaattikoetta varten, sillä se on jo huomenna. Aloitin lukemalla lisää integraatioista ja turvallisuusrooleista. Eilisen harjoituskokeiden jälkeen materiaali oli paljon selkeämpää luettavaa ja asiat painuivat paremmin mieleeni. Nyt esimerkiksi aineisto integraatiosta oli selkeämpää, sillä erotin jo termejä toisistaan ja osasin liittää näiden alle lisää termejä. Sain luotua mielessäni eräänlaista karttaa toiminnoista, joka auttaa, sillä opin parhaiten visuaalisesti.

Kun olin lukenut tarpeeksi, lähdin taas suorittamaan harjoituskoetta. Harjoituskoe oli paljon selkeämpi tällä kertaa, kun eilinen oli mielessäni ja ymmärsin kysymyksiä paremmin. Huomasin jopa osaavani vastata paremmin integraatiokysymyksiin ja rooleihin, mutta kysymykset, joissa minun täytyy laittaa prosesseja oikeaan järjestykseen, se on edelleen vaikeaa. Vastausten on mentävä täysin oikein kaikilta osilta, tai niistä ei saa ollenkaan pisteitä. Tällä kertaa pääsin tulokseen 56 % oikein. Huomasin jo muistavani kysymyksiä eiliseltä, ja vaikka niissä oli eri arvoja, se auttoi miettimään oikeaa vastausta. Kokeen lopussa saa myös jaon osa-alueittain osaamisestani ja tällä kertaa se näyttää paljon tasaisemmalta. Suurin osa osa-alueista oli 50 % alueella. Koodaaminen oli korkeammalla ja esimerkiksi integraatio vielä hiukan alempana.

Tulos on vielä kaukana vaadittavasta 80 % tuloksesta, enkä osaa sanoa millaisia kysymyksiä virallisessa kokeessa on, mutta oma tulokseni tuntuu jo ihan hyvältä. En ole lähdössä viralliseen testiin täysin ilman valmistautumista, vaan tiedän tehneeni töitä ja pärjääväni ainakin jotenkin. Työnantaja ei oleta, että kukaan pääsisi sertifikaattikokeita ensimmäisellä kerralla läpi ja mahdollisuuksia yrittää on niin paljon kuin haluaa. Yrityksemme on kuitenkin Microsoftin top 10 yhteistyökumppaneita, joten arvostus sertifikaatteja kohtaan on suuri. Niillä pystymme osoittamaan oman pätevyytemme

niin työntekijöinä, kuin yrityksenä ja Microsoft pitää meitä jatkossakin tärkeänä yhteistyökumppanina. Jäin vielä lukemaan aineistoa, harjoittelemaan tenttiä varten ja testasin ympäristöjen toiminnan. Tämän jälkeen olen valmis huomiseen sertifiikaattikokeeseen.

Keskiviikko 25.5.2022

Keskiviikko alkoi taas perinteisellä tiimipalaverilla ja keskustelimme siitä, miten koulutuksessani viimeinen tehtävä jäi tarkistamatta, sillä kouluttajani sairastui. Siitä oli jo noin kolme viikkoa ja kouluttajani oli edelleen kipeänä, mutta eräs vanhempi koodaaja tiimistäni halusi tutkia tehtävää kanssani. Sovimme ajan iltapäivälle, mutta tärkeimpänä minulla oli tänään sertifiikaattikokeen tekeminen. Voisimme katsoa tehtävää sen jälkeen.

Katoin aamusta ympäristön tietokannan importointia toisen aloittavan työkaverini kanssa ja osasin antaa hänelle neuvoa siihen. Latasimme tietokannan LSC:stä ja laitoimme sen rakentumaan. Tämän jälkeen lähdin valmistelemaan neuvottelutilaa, jonka olin varannut sertifiikaattikokeeseen. Sertifiikaatin tekemistä valvotaan jatkuvasti video- ja ääniyhteydellä, eikä huoneessa saa olla mitään häiritsevää. Pyyhin neuvotteluhuoneen taululta kaiken, teippasin oveen lapun, otin television irti virtalähteestä ja siirsin kaikki irtaimistot huoneen nurkkaan. Näin huoneen pöytä oli täysin tyhjä, eikä mitään häiriötä ole ympärillä.

Testasin vielä kertaalleen järjestelmän toimivuuden ja kun oli kokeen aika, kirjauduin sisään linkistä. Kaikki testaukset ja tehtävä vielä uudestaan ja ympäristöstä, sekä itsestään on otettava kuvat järjestelmään. Sen jälkeen jäin odottamaan vuoroani. Hetken odoteltua joku järjestelmän työntekijöistä otti yhteyttä äänipuhelulla. Ensimmäisenä hän pyysi minua poistamaan kuulokkeet päästäni ja sen jälkeen näyttämään tilan kameralla. Kaikki oli hyvin ja sain aloittaa kokeen tekemisen.

Kokeessa oli heti aluksi tapauskohtaisia kysymyksiä. Siinä oli esitelty yksi tapaus ja siihen liittyi kaikki seuraavat kuusi kysymystä. Tämän jälkeen tuli normaalimpia monivalintakysymyksiä ja lopuksi vielä oikein väärin väittämiä. Aikaa oli kaksi tuntia, eikä kokeessa ollut taukoja. Huomasin väsyväni paljon kokeen aikana jännityksestä ja kokeen vaikeudesta. Sain suoritettua kokeen aikarajan sisällä ja sain lopuksi pistesaldoksi 520/1000. 750 oli vaadittava raja, joten jäin kuitenkin siitä aika kauaksi, mutta harjoittelusta oli selkeästi ollut hyötyä. Taas olin vahvoilla ja heikoilla samoissa asioissa, kuin harjoittelussakin. Totesin, että tämä koe oli hyvä harjoitus, mutta minulla ei ole kiirettä uusilla asioilla. Tiesin taas asioita, joiden kanssa oli työskennellyt, joten on vain parempi jatkaa töitä ja saada kokemusta sen kautta. Vasta tämän jälkeen lähdin suorittamaan koetta uudestaan.

Kokeen jälkeen katsoimme kollegani kanssa viimeistä harjoitustehtävääni. Olin saanut toiminnon valmiiksi, mutta halusin itsekin tietää, että olenko toiminut oikein. Kollegani myös mainitsi, että teh-

tävän osaaminen ei ole itsestäänselvyys, sillä edes kaikki tiimistämme eivät ole tehneet integraatiotoimintoja. Kun osaan tämän, minulla on osaamista, jota ei kaikilta löydy. Esittelin sovellukseni kollegalleni ja toimintoa testataksemme lähetin postmanilla kutsun, joka sisälsi tallennettavat tiedot. Sain postmanilla koodin 200, joka viittasi hyväksytyyn kutsuun ja menin katsomaan tapahtumia FO:n puolelta. FO:ssa tiedot olivat myös menneet oikeaan tauluun, oikeilla tiedoilla. Kollegani oli myös tähän tyytyväinen ja kannusti minua esittelemään tuotoksen seuraavassa tiimipalaverissa. Muokkasimme koodia hieman selkeämmäksi ja siistimmäksi, mutta se oli muuten hyvä.

Torstai 26.5.2022

Helatorstai

Perjantai 27.05.2022

Tänään on viimeinen päiväkirjan seurantapäivä. Perjantai ei alkanut normaaliin tapaan tiimikokouksella, sillä meillä on myöhemmin tänään yrityksen omat kesäjuhlat. Säästimme siis aikaa varsinaisesta työpäivästä oikealle tekemiselle.

Minulle oli luotu uusi virtuaalikone uuteen asiakasprojektiin ja kaikki oikeudet näyttivät olevan kunnossa. Sain siis käynnistettyä koneen ja otettua siihen yhteyden. Tämän jälkeen kaikki oli tuttua aiemmasta virtuaalikoneen alustamisesta. Päivitin järjestelmän ja sovellukset, hain tietokannan ja aloin purkamaan sitä. SQL Serverillä alustin tietokannan ja varmistin, että se löytyy oikeasta paikasta. Tämän jälkeen tein muutoksen Visual Studioon ja yhdistin tietokannan sinne. Visual Studiosta yhdistin DevOpsin versiohallintaan, linkitin repos kansion models kansioon, rakensin mallit, synkronoin tietokannan ja hain uusimmat muutokset versiohallinnasta. Tämän jälkeen AOT rakentui oikein. Nyt ympäristö oli valmis projektien aloittamiseksi. Aiemmin oli puhetta, että eräs tiketti olisi entiteetin muokkaamista ja alan varmaan sillä liikkeelle. Koko päiväni kului ympäristön alustamiseen, joten muutoksen aloittaminen jäisi ensi viikolle.

Illasta meillä oli yrityksen kesäjuhlat, joka tavallaan merkkasi minulle myös opinnäytetyön tekemisen loppumista. Tämänpäiväinen tekeminen oli kuin suoraan aiemmin kirjoittamistani merkinnöistä, mutta tällä kertaa se sujui ilman ongelmia tai liikaa miettimistä. Aiemmista merkinnöistä ja pohdinoista on selkeästi ollut hyötyä ja olen sisäistänyt aiheita paremmin. Ensi viikolla jatkan asiakasprojektissa ja teen sitä, mitä työni varsinaisesti tulee olemaan.

Viikkoanalyysi

Tämä viikko oli viimeinen seurantaviikkoni päiväkirjan tekemisessä. Odotin, että olisin päässyt enemmän asiakasprojektien pariin, mutta harmikseni nykyisellä asiakkaalla ei ollut uusia töitä. Tämän takia viikko meni odotteluksi ja lopulta sain toisen asiakasprojektin itselleni. Tämän jälkeen voin suorittaa kumman vain ympäristössä muutoksia.

Sain kuitenkin tehtyä yhden tavoitteistani, joka oli sertifiikaattikokeen suorittaminen. Kyseessä oli ensimmäinen sertifiikaattikoe, jota ikinä kokeilin. Harjoittelin sitä varten paljon, jotta osaisin asian ja jotta saisin siitä enemmän irti. Lopulta en päässyt kokeesta läpi, mutta sain suhteellisen hyvän tuloksen ja painoin mieleeni paljon kokeen rakenteesta ja kysymysten tyypeistä. Lähdän suorittamaan koetta uusiksi ehkä noin puolen vuoden päästä, jolloin minulla on paljon enemmän kokemusta varsinaisesta FO kehittämisestä. Opin asiat paljon paremmin tekemällä ja visuaalisesti oppimalla, kuin teoriaa lukemalla.

Nyt olen virallisesti suorittanut kaiken, jota työnantaja on minulta halunnut koulutustani varten. Olenhan nyt jo mukana asiakasprojekteissa kolmatta viikkoa eli varsinainen siirtyminen työtehtäviin on sujunut sulavasti. Kesää kohden on aina tilanne, että mitään uutta ei varsinaisesti aloiteta asiakkaiden kesälomien takia, joten tällä hetkellä työt sijoittuvat ylläpitoon ja pieniin muutoksiin. Pääsin onneksi osalliseksi toiseen asiakasprojektiin, joten tulevaisuudessa tehtäviä alkaa tulemaan ripeämmällä tahdilla. Minulle aukaistiin myös yksi erillinen kehittäjäympäristö, joka on meidän tiimimme sisäinen siltä varalta, että minulle opetettaisiin uusia teknologioita. Näistä teknologioista mielenkiintoisimpana pidin AOT-Browseria ja Docentric-laajennusta.

AOT-Browser on ollut automaattisesti FO:n aiemmassa versiossa AX:ssa. Nykyisin sitä ei ole tarjolla, vaan se täytyy asentaa ulkopuolisena laajennuksena. AOT-Browser tuo AOT-objektipuun haettavaksi myös FO:n sovellukseen. Tämä helpottaa siinä, että nähdäksesi objektien tietoja, sinun ei tarvitse olla kehittäjä ja mennä etsimään tietoa Visual Studiosta. Voit esimerkiksi suoraan sovelluksesta painaa hiiren oikealla painikkeesta ja aukaista sen tiedot AOT-näkymässä. Tästä näet suoraan objektin määrittäykset ja pystyt jopa tarkastelemaan lähdekoodia. Tämän on kehittäjälle hyödyllistä, että sinun ei tarvitse aina debuggerilla paikantaa kaikkea, vaan saat suuntaa antavan avustuksen suoraan sovelluksesta. (Woods 2021).

Docentric on laajennus, joka helpottaa raporttien tekemistä. Sen avulla raporttien ulkoasua on helpompaa muokata, tulosteiden printtaaminen ja lähettäminen on helpompaa, sekä tietojen täyttämisen ja muokkaaminen onnistuu valikoista. Docentric tekee tulosteiden arkistoinnista myös kevyempää. Yrityksemme käyttää Docentric-laajennusta todella monissa asiakasprojekteissa ja osajia sen käyttämiseen on vain muutamia. Tämänkin takia tietoa on jaettava ja sen käyttö opetetaan minulle myös tulevaisuudessa. (Docentric 2022).

Kun kouluttaudun useamman toiminnon osaajaksi, saan myös samanaikaisesti useamman mentorin. Alyssa Westringin (2021) mukaan useampi mentori on hyödyksi urakehitykselle. Hänen mukaansa ongelma mentorissa on se, että niin hyvässä kuin huonossa, seuraat täysin mentorisi askeleita. Teet pakostakin samoja valintoja, mitä hän on tehnyt ja seuraat hänen uraansa. Jotta pystyisit refleктоimaan omaa oppimista ja tekemään paremmin omia valintoja, useampi mentori on tarpeen. On myös kohtuutonta kuvitella, että yhdellä mentorilla olisi vastaus kaikkiin kysymyksiin. Westringin mukaan sinun tulisi ajatella missä haluat olla urallasi viiden vuoden päästä, mitä taitoja haluat kerätä siihen mennessä ja mitä kontakteja haluat luoda. (Westring 2021).

Jos seuraisin yhtä mentoria, olisin hänen tietojensa, taitojen ja kontaktien varassa. Kannattavinta oman kehitykseni ja uran kannalta on siis kerätä taitoja laajasti ja hankkia useampi mentori. Työttömistä olen tehnyt yhteistyötä jo muutaman vanhemman työntekijän kanssa, ja he ovat sanoneet auttavansa minua. Uudessa asiakasprojektissa minulla on uusi mentori ja kun opettelen uusia teknologioita, saan taas uusia kouluttajia ja kontakteja. Koen siis urakehityksen olevan hyvässä kanti-
missa, ja että minulla on valtaa vaikuttaa oman kehitykseni suuntaan. Työni tulee olemaan jatkuvaa uuden oppimista ja kehittymistä ja olen mielestäni luonut hyvän alustan oppimiselle päiväkirjan avulla ja omalla tekemisellä.

4 Pohdinta ja päätelmät

Kahdeksan viikon seurantajakso tuntui hyvin nopealta ja opettavaiselta. Nyt kun katson alkutilannetta, asettamani tavoitteet ja lähtötilanteet tuntuvat todella kaukaisilta. Kahdeksan viikon kokonaiskainen työ näiden työkalujen kanssa, sekä aiheesta kirjoittaminen päiväkirjamuotoisesti on auttanut aiheiden sisäistämisessä valtavasti. Prosessin alussa pelkäsin haasteeksi sitä, että käsiteltävää ja tehtävää olisi liikaa ja ylikuormittuisin työtehtävistäni, enkä pystyisi antamaan tarpeeksi aikaa päiväkirjan tekemiseen. Nyt kuitenkin pystyn varmuudella toteamaan, että päiväkirjamaisen opinnäytetyön tekeminen uuden työn aloittamisen aikana on ollut oikea valinta itselleni.

Olen käynyt koulussa kurssin pilvipalveluista ja olen teorian avulla syventynyt niiden toimintaan, varsinkin Amazonin AWS-tarjontaan. Minulta kuitenkin uupui käytännön taidot, joten otin pilvipalveluihin tutustumisen yhdeksi päätavoitteista. Olen saanut seurantajakson aikana luoda omia palveluita omaan pilviympäristöön, sekä lopulta saanut käyttöoikeudet asiakkaan ympäristöihin. Seurantajakson aikana yllättävimpänä asiana pilvipalveluista nousi esiin Azure DevOps. Olin kuullut termistä aiemmin, mutta en perehtynyt siihen sen enempää. Tutustuin aiheeseen paljon omalla ajallani, tein omia projekteja testiympäristössä, tein DevOpsista esityksen kouluun ja liityin yrityksen DevOps-tiimiin. DevOpsin käytön ymmärtäminen on mielestäni tärkeää myös kehittäjille ja mitä enemmän perehdyin aiheeseen, sitä enemmän löysin siitä hyödyllisiä ominaisuuksia ja tietoa, joka auttoi omaa kehittymistäni.

Olen suorittanut koulussa ohjelmistotuotannon linjan opinnot, mutta FO-ympäristö ja koodikieli X++ olivat täysin tuntemattomia. Nämä ovat kuitenkin asiat, joita tulen käyttämään töissä päivittäin, joten valitsin ne toiseksi pääaiheeksi. FO-ympäristön oppiminen on mennyt aalloittain vaihtuen ”tämä on yksinkertaista” ja ”en tule ikinä oppimaan kaikkea” mietteiden välillä. Kuten mentorini aiheen ilmaisi: ”Tässä menee usein nakit ja muusit sekaisin”. Järjestelmän oppiminen ei varmasti tule loppumaan ikinä, mutta olen kuitenkin saanut hyvän peruskuvan kaikista elementeistä ja osaan etsiä tietoa. Tärkeimpänä työmenetelmänä minulle nousi esiin debuggaus. Ilman sitä ei tietojen hakemisesta tulisi mitään.

X++ koodikielenä on tullut tutuksi, mutta en usko, että senkään oppiminen loppuisi varsinaisesti koskaan. Koodaaminen on rajattua FO-ympäristön sisässä ja huomaan päivittäin löytäväni uusia tapoja, jolla saan suoritettua toimintoja. Tuntuu, että kaikki tieto kielestä on niin vähäistä, että kaikki täytyy havaita itse tai kysymällä muilta tekijöiltä. Olen kuitenkin omaksunut kielen perusperiaatteet hyvin ja tiedän, millaista koodia järjestelmään tehdään. Olen saanut myös hyviä vinkkejä mentoriltani, joista varsinkin se on pureutunut mieleeni, että en saa ikinä kovakoodata mitään arvoja. Seu-

raavana askeleena olen pohtinut ottavani kurssin C#-kielestä, sillä näkisin sen kehittävän ohjelmointitaitojani eniten. X++ kuitenkin pohjautuu C#:iin ja sisältää lopulta vain pieniä eroja, joten kurssien käyminen ja teorian lukeminen tukisi varmasti omaa tekemistäni töissä.

Asetin kolmanneksi tavoitteeksi tiimityöskentelyn oppimisen. En ole aiemmin työskennellyt IT-alalla, tiimityöskentely virallisissa töissä on minulle uutta ja etätö on tuonut omat muutoksensa näihin. Aloin käymään tiimipalavereissa seurantajakson puolesta välin ja olin virallisissa asiakasprojekteissa viimeisen kolme viikon ajan. Sain oppia käytäntöjä aluksi tiimiltä, mutta paras apu oli lopulta oma mentorini. Huomasin, että minun täytyy olla aktiivinen ja kysellä kysymyksiä, sekä myöntää etten osaa joitakin asioita. Vaikka tietäisin aiheesta vähän, huomasin usein, että mentori pystyi opettamaan siitä paljon enemmän. Tätä kautta otin aktiivisemmin muihin tiimin jäseniin yhteyttä ja kyselin asioiden edistymisestä. Nyt olen päässyt useampaan projektiin mukaan ja tätä kautta saan useampia mentoreja. Tiimityöskentelyn ja kommunikoinnin opettelu oli tärkeää, sillä hyvällä aktiivisuudella pystyn saamaan useampia mentoreja ja tätä kautta päättämään lopulta myös oman urani suuntaa.

Opinnäytetyön tekeminen on ollut kokonaisuudessaan minulle opettavainen kokemus ja koen ammatillisen osaamisen kehittyneen sen myötä suuresti. Kehitys uudessa työssä on väistämätöntä, varsinkin kun minua koulutettiin tehtävään, mutta opinnäytetyö on auttanut aukaisemaan ajatusiani. Päiväkirjan kirjoittaminen ja analyysien tekeminen on itseasiassa auttanut sisäistämään uuden työn paljon paremmin. Huomasin varsinkin raportoinnin alussa ja asiakasprojektien alkaessa, että ajatukseni alkoi käymään ylikierroksilla ja asiat alkoivat sekoittua. Kirjoitin päivän aikana jatkuvasti muistiinpanoja ylös, kirjoitin ne illalla päiväkirjaan ja viikkoanalyysiin päästäessä oppimani tuntuikin jo selkeältä. Sen opin, että aion jatkossakin jatkaa muistiinpanojen tekemistä ja luen lisätietoa nettilähteistä.

Olen jo usein palannut opinnäytetyön aiheisiin ja hakenut sieltä tietoa työtehtäviin. Olen huomannut selkeästi dokumentoinnista paljon hyötyä ja varsinkin kuvien tekeminen on ollut itselleni tärkeää. Olen visuaalinen oppija ja olen pystynyt töissä muistamaan useita prosesseja vain siksi, että olen tehnyt niistä kuvia opinnäytetyötä varten. Olen huomannut, että ohjelmoinnissa ja järjestelmiä käyttäessä opin myös parhaiten visualisoimalla asioita mielessäni. Sertifikaattikokeessakin osasin asiat, joita olin tehnyt ja aiemmin nähnyt. Aion jatkaa tulevaisuudessakin kuvien tekemistä tai etsimistä, sillä se on auttanut paljon.

Vaikka olen oppinut jo paljon, olen vieläkin työurani alussa. Työtehtävät tulevat tästä eteenpäin vain vaikeutumaan. Tämän takia koen suurena vahvuutena, että olen kirjoittanut opinnäytetyötä ja luonut itselleni vahvan pohjan, jolle lähden rakentamaan tätä uraa. Yhtiöstä ja omasta organisaas-

tiosta on tullut minulle tuttuja ja tunnen että olen saavuttanut heidän tavoitteensa koulutukseni suhteen. Olen oppinut hyviä työskentelymalleja, olen saavuttanut asettamani tavoitteet ja olen valmis oppimaan lisää.

Lähteet

Burak, A. 2021. How to Become a Good Software Engineer Mentor: 10 Tips. Luettavissa: <https://relevant.software/blog/good-software-engineer-mentor-tips/>. Luettu: 27.05.2022

Codepipes, 2018. Software Testing Anti-patterns. Luettavissa: <http://blog.codepipes.com/testing/software-testing-antipatterns.html>. Luettu: 04.04.2022

Docentric, 2022. Create Better Repors in MS Dynamics 365 for Finance and Operations. Luettavissa: <https://ax.docentric.com/>. Luettu: 27.05.2022

Fox, S. 2022. Dynamics 365 Business Central vs Finance & Supply Chain Management: How to choose between the two. Luettavissa: <https://www.columbusglobal.com/en-gb/blog/dynamics-365-business-central-vs-finance-supply-chain-management-how-to-choose-between-the-two>. Luettu: 14.04.2022

Lee, J. 2022. Why You Should Avoid Hard Coding and Three Alternative Solutions. Luettavissa: <https://admin.salesforce.com/blog/2021/why-you-should-avoid-hard-coding-and-three-alternative-solutions>. Luettu: 21.05.2022

Marrocos, M. 2020. An Overview of Azure Integration Services. Luettavissa: <https://medium.com/devops-cloud-it-career/an-overview-of-azure-integration-services-fe291603cd34>. Luettu: 30.04.2022

Microsoft 2018a. AOT Overview. Luettavissa: <https://docs.microsoft.com/en-us/dynamicsax-2012/developer/aot-overview#:~:text=In%20Microsoft%20Dynamics%20AX%2C%20the,%2C%20forms%2C%20and%20so%20on>. Luettu: 09.04.2022

Microsoft, 2018b. How to: Create Queries by Using X++. Luettavissa: <https://docs.microsoft.com/en-us/dynamicsax-2012/developer/how-to-create-queries-by-using-x>. Luettu: 01.05.2022

Microsoft, 2021. X++ and C# comparison. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/dev-ref/xpp-cs-comparison>. Luettu: 17.04.2022

Microsoft, 2022a. Finance and Operations application documentation. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/fin-ops/>. Luettu: 04.04.2022

Microsoft, 2022b. Microsoft Learn. Luettavissa: <https://docs.microsoft.com/fi-fi/learn/>. Luettu: 04.04.2022

Microsoft, 2022c. Exam MB-500: Microsoft Dynamics 365: Finance and Operations Apps Developer. Luettavissa: <https://docs.microsoft.com/en-us/learn/paths/introduction-develop-finance-operations/>. Luettu: 07.04.2022

Microsoft, 2022d. Cost accounting home page. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/finance/cost-accounting/cost-accounting-home-page>. Luettu: 14.04.2022

Microsoft, 2022e. Advanced filtering and query syntax. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/fin-ops/get-started/advanced-filtering-query-options>. Luettu: 01.05.2022

Microsoft, 2022f. Integration between Finance and Operations apps and third-party services. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/data-entities/integration-overview>. Luettu: 08.05.2022

Microsoft, 2022g. Set up Azure DevOps code sharing and create a build pipeline. Luettavissa: <https://docs.microsoft.com/en-us/dynamics365/commerce/e-commerce-extensibility/set-up-code-sharing-build-pipeline>. Luettu: 08.05.2022

Microsoft, 2022h. Get started with AzCopy. Luettavissa: <https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10>. Luettu: 14.05.2022

Microsoft, 2022i. Quickstart: Import a BACPAC file to a database in Azure SQL Database or Azure SQL Managed Instance. Luettavissa: <https://docs.microsoft.com/en-us/azure/azure-sql/database/database-import?view=azuresql&tabs=azure-powershell>. Luettu: 14.05.2022

Mäntyneva, M. 2016. Hallittu projekti: Jäntevästä suunnittelusta menestykselliseen toteutukseen. 1. painos. Kauppakamari. Helsinki.

Ramer, P. 2021. How to create a Form in D365. Luettavissa: <https://dynamics365musing.com/how-to-create-a-form-in-d365/#:~:text=Right%20click%20on%20your%20project%2C%20and%20select%20Add%3ENew%20Item,a%20name%20for%20the%20form>. Luettu: 24.04.2022

Ryhänen, J. 2022. Azure integration Services: Azure integraatioalustana. Luettavissa: <https://www.devisioona.fi/2021/05/azure-integration-services-azure-integraatioalustana>. Luettu: 30.04.2022

Lee, J. 2022. Why You Should Avoid Hard Coding and Three Alternative Solutions. Luettavissa: <https://admin.salesforce.com/blog/2021/why-you-should-avoid-hard-coding-and-three-alternative-solutions>. Luettu: 21.05.2022

Tawde, S. 2022. What is Debugging? Luettavissa: <https://www.educba.com/what-is-debugging/>.
Luettu: 21.05.2022

Telerik, 2022. Arrange Act Assert. Luettavissa: <https://docs.telerik.com/devtools/justmock/basic-usage/arrange-act-assert>. Luettu: 04.04.2022

Westring, A. 2021. Why You Need Multiple Mentors. Luettavissa: <https://hbr.org/2021/06/why-you-need-multiple-mentors> Luettu: 27.05.2022

Woods, T. 2021. AOT Browser: D365 Finance and Operations. Luettavissa: <https://dynamics-tips.com/aot-browser-d365-finance-and-operations/> Luettu: 27.05.2022