

Perttu Moilanen

Uuden työntekijän perehdytys IT-alan ohjelmointitehtäviin

Päiväkirja integraatiokehittäjän näkökulmasta

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutusohjelma
Tammikuu 2020**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Toukokuu 2020	Tekijä Perttu Moilanen
Koulutusohjelma Tieto- ja viestintätekniikka		
Työn nimi UUDEN TYÖNTEKIJÄN PEREHDYTYKSEN IT-ALAN OHJELMOINTITEHTÄVIIN. Päiväkirja integraatiokehittäjän näkökulmasta.		
Työn ohjaaja Sakari Männistö, Helena Åkerlund		Sivumäärä 45
Työelämäohjaaja Juho Muuraiskangas		
<p>Nykyään monissa yrityksissä käytetään paljon erilaisia järjestelmiä ja ohjelmistoja, joiden yhteentoimivuus on tärkeää. Haasteita aiheuttaa tiedon hajaantuminen useisiin eri järjestelmiin ja tiedostomuotoihin, jolloin yhdestä järjestelmästä saatava tieto ei välttämättä ole luettavissa toisella tai tietoa joudutaan siirtämään ja koostamaan työläästi käsityönä. Integraatioiden avulla päivittäinen työnteko muuttuu sujuvammaksi ja niiden avulla on myös mahdollista saavuttaa merkittäviä säästöjä.</p> <p>Integraatiolla tarkoitetaan eri tavoin toteutettujen ohjelmistojen ja järjestelmien liittämistä toisiinsa siten, että kyseiset osat toimivat paremmin keskenään. Tietojärjestelmäintegraatiossa erilliset järjestelmät, ohjelmistot ja laitealustat yhdistetään yhtenäiseksi kokonaisuudeksi. Alfamella kyse ei kuitenkaan ole vain järjestelmien toisiinsa kytkemisestä, vaan prosessiohjatulla integraatioilla rakennetaan kokonaisuuksia, joiden avulla asiakkaat automatisoivat ja kehittävät liiketoimintaansa. Mikropalveluarkkitehtuurin periaatteita noudattaen mahdollistetaan nykyaikaiset API-ratkaisut, joissa sovelluskokonaisuus on hajautettu ja jaettu itsenäisiksi prosesseiksi, usein eri palvelimille tai pilvipalveluun.</p> <p>Tässä opinnäytetyössä kuvaan päiväkirjan muodossa edistymistäni osana integraatioita laativaa tiimiä. Opin harjoittelun aikana erilaisia integraatioprojekteissa käytettäviä tekniikoita ja työskentelytapoja. Opinnäytetyössä seurataan työskentelyäni kymmenen viikon ajan. Kerron päivittäisen työskentelyni aikana sattuneista tärkeimmistä tapahtumista ja sitä kautta kertyneistä opituista asioista.</p>		
Asiasanat API, Activiti, Container, Hashmap, Java, JSON, Log, Mule, Xml		

ABSTRACT

Centria University of Applied Sciences	Date May 2020	Author Perttu Moilanen
Degree programme Information and communication technology		
Name of thesis Familiarization of new worker in IT field of programming. Diary from the perspective of an integration programmer.		
Instructor Sakari Männistö, Helena Åkerlund		Pages 45
Supervisor Juho Muuraiskangas		
<p>Abstract</p> <p>Nowadays many businesses use a lot of different systems and applications that need to be compatible with each other. Challenges are caused by fragmentation of information into various systems and file formats, which occurs when the information from one system may not be readable by another system or the information needs to be transferred and gathered laboriously by manual work. With integrations, the daily work will turn more fluent and it is possible to achieve remarkable savings.</p> <p>Integrations means combining differently implemented software and systems into each other so that those items work better together. In information system integration, the separate systems, applications and hardware platforms are combined into a single entity. At Alfame the intention is not just to combine systems together but to use process guided integrations to build entities with which the customers can automate and develop their business. Using principles of Micro Services Architecture modern API-solutions are made possible, whereby the collection of applications is distributed and divided into independent processes, often into various servers or into a cloud service.</p> <p>In this thesis I am using a diary format to illustrate my progress as a part of an integrations implementation team. During my internship I learned different kinds of techniques and ways of working. The diary follows my work for the duration of ten weeks. I tell about the most important events and about the things I learned through that process.</p>		
<p>Key words API, Activiti, Container, Hashmap, Java, JSON, Log, Mule, Xml</p>		

KÄSITTEIDEN MÄÄRITTELY

API	(Application Programming Interface) Ohjelmointirajapinta tai määritelmä, jonka mukaisesti erilaisten ohjelmien ja järjestelmien välillä voidaan siirtää tietoa. (Wikipedia. 2020. Ohjelmointirajapinta)
Anypoint	Anypoint on Eclipse-johdannainen IDE, jota Mulesoft tarjoaa osana Anypoint Platformia, johon kuuluu Mule ESB ja Anypoint IDE yhdessä paketissa.
Activiti	Työnkulun ja liiketoimintaprosessien hallintajärjestelmä, jonka avulla prosesseja määritellään ja suoritetaan.
CRM	(Customer Relationship Management) Asiakassuhteiden hallintaan liittyvä järjestelmä, esim Dynamics.
CMS	(Content Management System) Sisällönhallintaan liittyvä järjestelmä, esim Contentful.
Contentful	Sisällönhallintajärjestelmä sovelluksille ja web-sivuille.
Docker	Työkalu, jonka avulla ohjelmistoja ja niiden tarvitsemia muita komponentteja asennetaan itsenäiseen konttiin, joka voidaan suorittaa halutulla palvelimella.
Dynamics	Järjestelmä, jota voidaan käyttää palvelimella resurssien ja asiakassuhteiden hallintaan.
IDE	(Integrated Development Environment) Työkalu, jota käytetään varsinaisen ohjelmakoodin kirjoittamiseen ja kääntämiseen.
Intellij Idea	IDE, jonka kanssa voidaan luoda myös Mule-sovelluksia käyttäen Java, Maven ja Spring teknologioita yhdessä Mule ESB pluginin kanssa.
Java	Luokkiin ja olio-ohjelmointiin perustuva yleiskäyttöinen ohjelmointikieli.
JSON	(JavaScript Object Notation) tiedoston muoto, jossa avain-arvopareista muodostettua tietoa jaotellaan ja ryhmitellään aaltosulkujen avulla.
Kontti	Dockerilla luotu virtuaalikonetta muistuttava tapa paketoita ja asentaa ohjelmistoja eräänlaiseen säiliöön, josta niitä on myöhemmin helppo suorittaa aina samalla tavalla eri palvelinympäristöissä. Konttiin paketoidaan mukaan halutun ohjelman lisäksi kaikki ne muut ohjelmat ja komponentit, joista sen toiminta riippuu.
Log	Ohjelman suorituksen aikana kerätään ohjelman toimintaan liittyvää tietoa, jota voidaan hyödyntää esim. poikkeustilanteiden selvittämisessä.
Logitus	Varsinainen toiminnallisuus ja logiikka, jolla log-tieto kerätään.
Map	Tietorakenteita, jossa tieto muodostetaan avain – arvo pareista, joissa jokaisella arvolla on oma uniikki avain. Map-rakenteita ovat esimerkiksi Hashmap ja LinkedHashMap.
Mule	MuleSoft nimisen yrityksen kehittämä Mule Enterprise Service Bus on kevyt Javapohjainen integraatioalusta.

Mule flow	Mule-ohjelmat käsittelevät tietoa Mule komponenttien avulla, jotka asetetaan yhteen tai useampaan vuokaavioon, jossa komponentit käsitellään halutussa järjestyksessä. (Mulesoft LLC. Flows and Subflows)
Parseri	Jäsennin, jonka avulla voidaan poimia halutut tiedot tietyn tietorakenteen mukaisesta tiedosta.
Scrum	Päivittäinen kokous, jossa kerrataan mitä on eilen tehty ja suunnitellaan, mitä tänään tehtäisiin ja käydään läpi mahdolliset haasteet ja esteet toiminnalle. Scrumia käytetään myös yleisesti nimityksenä koko ketterän ohjelmistokehityksen toimintamallista, jossa Scrum-palaverit ovat yhtenä osana päivittäistä toimintaa.
Sprintti	Muutaman viikon jakso Scrum-työskentelyä, jonka aikana saadaan tuotettua työn alla olleeseen tuotteeseen ratkaisu tai parannus.
Stub	Palvelimen korvaava osa ohjelmistoa. Tätä käytetään testitilanteessa esim. silloin kun halutaan testata toistuvasti tiedon käsittelyä ohjelman suorituksen aikana ja tiedon hakemiseksi ei haluta joka kerta hakea tietoa palvelimelta vaan ohjelmalle annetaan vakiovastaus, joka on saman muotoinen kuin palvelimelta tuleva tieto olisi. Tällöin on myös mahdollista testata tiedon käsittelyä ilman verkkoyhteyksiä.
Yksikkötesti	Ohjelmoinnissa käytettävä yksi tai useampi Java-luokka, jolla testataan ohjelman toimintaa tekemällä oletuksia siitä, millaisia paluuarvoja ohjelman pitäisi käyttäjälle antaa ja vertaamalla näitä oletuksia todellisiin ohjelmalta saatuihin arvoihin.
XML	(eXtensible Markup Language) tiedoston muoto, jossa tietoa ryhmitellään hakasulkujen avulla.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
1.1 Työn tavoite	1
1.2 Alfame Systems Oy	1
2 LÄHTÖTILANTEEN KUVAUS	3
2.1 Oman nykyisen työn analyysi	3
2.2 Sidosryhmät työpaikalla	3
2.3 Vuorovaikutustaidot työpaikalla.....	5
3 PÄIVÄKIRJARAPORTOINTI	5
Seurantaviikko 1	5
Maanantai.....	5
Tiistai.....	6
Keskiviikko	6
Torstai	6
Perjantai.....	7
Viikkoraportti	7
Seurantaviikko 2	8
Maanantai.....	8
Tiistai.....	9
Keskiviikko	9
Torstai	10
Perjantai.....	11
Seurantaviikko 3	12
Maanantai.....	12
Tiistai.....	12
Keskiviikko	13
Torstai	13
Perjantai.....	14
Viikkoraportti	14
Seurantaviikko 4	15
Maanantai.....	15
Tiistai.....	16
Keskiviikko	16
Torstai	17
Perjantai.....	18
Viikkoraportti	18
Seurantaviikko 5	19
Maanantai.....	19
Tiistai.....	19
Keskiviikko	20
Torstai	20
Perjantai.....	21
Viikkoraportti	22
Seurantaviikko 6	22

Maanantai.....	22
Tiistai.....	23
Keskiviikko	23
Torstai	24
Perjantai.....	24
Viikkoraportti	25
Seurantaviikko 7	25
Maanantai.....	25
Tiistai.....	26
Keskiviikko	26
Torstai	27
Perjantai.....	28
Viikkoraportti	28
Seurantaviikko 8	29
Maanantai.....	29
Tiistai.....	29
Keskiviikko	30
Torstai	30
Viikkoraportti	32
Seurantaviikko 9	32
Maanantai.....	32
Tiistai.....	33
Keskiviikko	34
Torstai	34
Perjantai.....	34
Viikkoraportti	35
Seurantaviikko 10	36
Maanantai.....	35
Tiistai.....	36
Keskiviikko	37
Torstai	37
Perjantai.....	38
Viikkoraportti	39
4 POHDINTA	40
LÄHTEET	38
LIITTEET.....	39

1 JOHDANTO

1.1 Työn tavoite

Tämän opinnäytetyön aiheena on kouluttautumiskurssini uutena työntekijänä Alfame Systems Oy:n Kokkolan toimipisteen integraatioheimossa. Opinnäytetyössä kuvataan päiväkirjamuotoisesti työhöni liittyviä tapahtumia ja tehtäviä ja pulmatilanteita, joita yritän ratkaista. Kestoltaan tämä päiväkirja kattaa noin kymmenen viikon ajanjakson helmikuun toiselta viikolta eteenpäin. Viikoittain teen yhteenvedon tärkeimmistä asioista, joita viikon aikana on tapahtunut, ja siitä, mitä olen oppinut.

Työhön perehtymisessä tarvitaan ennen kaikkea kykyä oppia uusia asioita ja tekniikoita nopeasti. Javaohjelmointitaidot ovat toki oleellisia, mutta Java on vain tukijalka, jonka päälle rakennetaan muun muassa Activiti-, Mule-, Spring- ja Maven-tekniikoiden avulla varsinainen kehitysympäristö. Tärkeimpänä tavoitteena minulla on oppia riittävä määrä tekniikoita ja saada rutiinia työssä käytettäviin työkaluihin, jotta voin pitää työllistymismahdollisuuksiani vähintäänkin kohtuullisina. Yhtenä mittarina tässä voi käyttää sitä, että pystyisin tämän 10 viikon lopussa työskentelemään pitkiä jaksoja itsenäisesti kysymättä jatkuvasti apua kollegoilta.

1.2 Alfame Systems Oy

Alfame on vuonna 2004 perustettu yksityinen informaatioteknologian alan yritys, joka on erikoistunut tuottamaan integraatoratkaisuja asiakasyrityksilleen. Alfamen visiona on olla tunnetusti Suomen luotetuin API- ja integraatoratkaisuiden kehittäjä ja toimittaja. Alfamen asiakkaat ovat nykyään suurimmaksi osaksi pääkaupunkiseudulta, mutta yritys on syntynyt ja kasvanut Keski-Pohjanmaalla. (Alfame 2020b. Kotisivu.)

Yksinkertaisimmillaan integraatio voi olla esimerkiksi tiedonsiirtoa kahden eri ohjelmiston välillä samalla laitteella, mutta toisaalta laajemmissa kokonaisuuksissa toteutetaan yhteenliittymiä useiden eri järjestelmien välillä, jotka voivat sijaita hyvinkin kaukana toisistaan. (Alfame 2020a. Integraatio-opas.)

Asiakasorganisaatioissa tarve integrointiin voidaan havaita, kun käytössä olevasta järjestelmästä puuttuu tärkeitä ominaisuuksia, jotka löytyisivät jostain muusta järjestelmästä. Toisen järjestelmän tarjoamia

tietoja ei kuitenkaan aina voida tuoda muihin järjestelmiin vaan on käytävä läpi useita eri järjestelmiä yksi kerrallaan. Integraation avulla hajaantunut tieto saadaan yhdistettyä haluttuun muotoon niin, että päivittäinen työ helpottuu ja siitä voidaan poistaa turhia toistuvia rutiineja. (Alfame 2020a. Integraatio-opas.)

Tyypillisesti tiedon siirtyminen esimerkiksi toiminnanohjausjärjestelmästä laskutusjärjestelmään on liiketoiminnalle kriittistä. Samoin asiakkuudenhallinta-järjestelmien integraatio muihin järjestelmiin on oleellista, jotta asiakassuhteiden ylläpito onnistuu sujuvasti. Ajoittain halutaan myös yhdistää useita vanhoja järjestelmiä yhdeksi uudeksi järjestelmäksi, johon kehitetään kaikki asiakkaan toivomat ominaisuudet. Tällöin haluttu järjestelmä kehitetään asiakkaan toiveita vastaavaksi. (Alfame 2020a. Integraatio-opas.)

Integraatiolla saavutettavia liiketoiminnallisia hyötyjä ovat muun muassa manuaalisen työn ja siitä aiheutuvien virheiden väheneminen automatisoinnin myötä, tehostunut yhteistyö eri sidosryhmien välillä, tietojärjestelmäarkkitehtuurin yksinkertaistuminen ja palvelutehokkuuden parantuminen: Esimerkiksi tilanteet, joissa voi syntyä inhimillisiä virheitä, kuten näppäilyvirheitä, vähentyvät automatisoinnin myötä ja tiedonsiirto alihankkijoiden ja tavarantoimittajien välillä helpottuu, kun toiminnanohjausjärjestelmä ilmoittaa saapuvasta tavarasta ja lähettää vastaanotosta kuittauksen toimittajalle. (Alfame 2020a. Integraatio-opas.)

Onnistunut integraatio toteutuu liiketoiminnan kipupisteiden ja tavoitetilojen määrittelyn kautta. Pelkästään tiedon siirtäminen paikasta toiseen ei riitä vaan myös liiketoiminnalliset tarpeet tulee huomioida. Tätä varten integraation toteuttajan on saatava riittävästi tietoa liiketoiminnasta ja sen prosesseista. Integraatioprojektin alussa suoritetaan aina liiketoimintakatsaus, joka säästää lopulta aikaa, vaikka usein haluttaisiin perehtyä saman tien projektin tekniseen toteutukseen. (Alfame 2020a. Integraatio-opas)

2 LÄHTÖTILANTEEN KUVAUS

2.1 Oman nykyisen työn analyysi

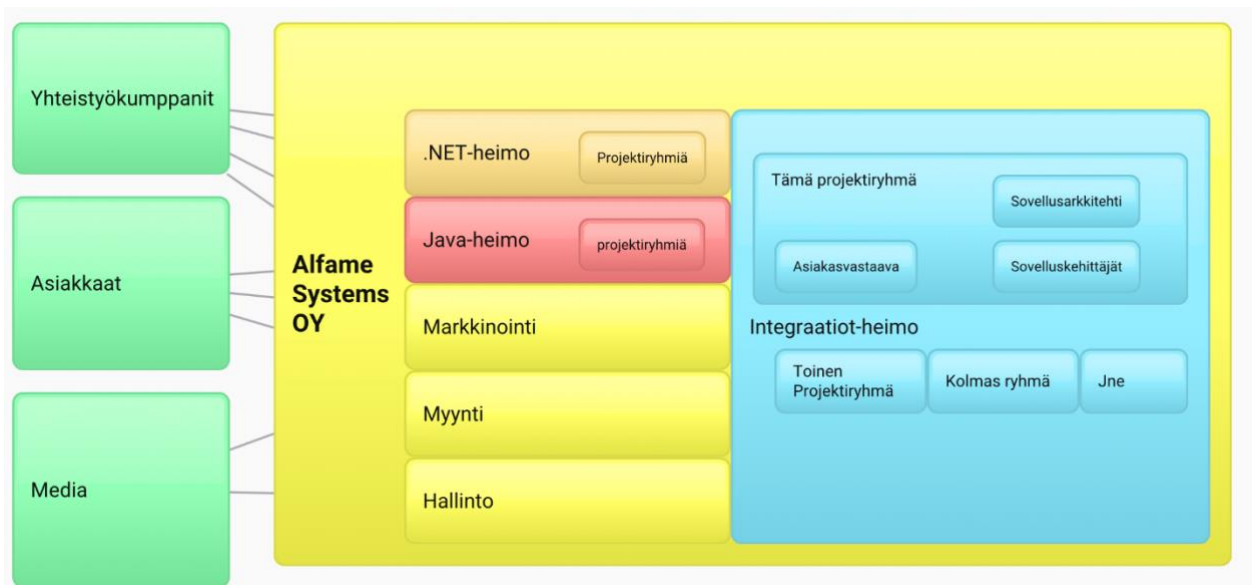
Alan kirjoittamaan päiväkirjaa tilanteesta, jossa olen opintojeni loppusuoralla ja viimeinen kevätlukukausi on alkamassa. Opiskelin ensin ammattikorkeakouluopintojen lisäksi noin kuukauden verran Java-ohjelmointia verkkokurssilla. (Buchalka, T. Lochert, G. 2019. Java Programming Masterclass for Software Developers.)

Lisäksi olen nyt toiminut harjoittelijana Alfamella noin kuukauden ajan. Tuosta ajasta kaksi viikkoa olen opiskellut Mulesoftin ja Udemyn sivuilta löytyvien oppaiden avulla Mule-flowien tekemistä ja Ansiblea. Lisäksi vajaan kahden viikon ajan olen ollut mukana asiakasprojektissa, jossa työstimme integraatiosovelluksia. Minun tehtäviini kuuluu integraatiosovellusten toteuttaminen käyttäen Mulesoft Mule ESB -integraatioalustaa. Työhön kuuluu ohjelmointia ja sovellusten testaamista. Työssä tarvitaan loogista päättelykykyä ja uusien tekniikoiden ja työkalujen oppimiskykyä ja -halua. Työn aloittamiseen tarvitaan perustason ohjelmointitaidot, tiedon etsintätaitoja ja kommunikointitaitoja, jotta voi kysyä ja oppia lisää kollegoilta. Tähän mennessä olen työharjoittelussa jo oppinut Mule ESB:n perusteet ja kasvattanut Java-ohjelmointitaitojani.

Tämänhetkinen taitotasoni on aloittelevan toimijan taso. Tarvitsen vielä säännöllisesti kollegoiden apua työssä selviytymiseen. Kysyn ja saan apua kollegoilta Chatissa ja välillä myös pyytämällä kollegaa saapumaan fyysisesti työpisteelleni neuvomaan minua. Ajoittain saan kuitenkin osia tehtävistä tehtyä itsenäisesti tietoa etsien. Ammatillisessa kehittämisessäni olen vielä alkutaipaleella, mutta toivon harjoittelun aikana pääseväni sellaiselle tasolle, jolla pääsen aloittamaan työskentelyn heti kesällä valmistuttuani. Vaikka en vielä osaa enkä tiedä paljonkaan integraatioista ja siinä käytettävistä työkaluista, tunnen silti pystyväni kehittymään tässä työssä vaadittavalle tasolle muutamassa kuukaudessa. Tässä vaiheessa tärkeintä on saada kokemusta erilaisista sovelluksista ja tilanteista, joita niitä kehitettäessä voi tulla vastaan. Erilaisten virheilmoitusten tulkinta on yksi tällainen asia, jossa joudun aluksi kyselemään paljon kollegoilta, mutta tämä varmasti helpottuu kokemuksen karttuessa.

2.2 Sidosryhmät työpaikalla

Työskentelen Alfamella Integraatiot-heimossa kehitystiimissä, johon kuuluvat asiakasvastaava, sovellusarkkitehti ja kolme sovelluskehittäjää. Lisäksi keskustelen satunnaisesti heimovastaavan ja muiden työntekijöiden kanssa. Pääasiallisesti kuitenkin työskentelyni koskettaa tällä hetkellä tämän nimenomaisen projektitiimin tehtäviä. Eniten jollain tavalla omaa tekemistäni koskettava ulkoinen sidosryhmä on tämän integraatioprojektin asiakas. Alfamen sidosryhmät on kuvattu alla (KUVIO 1).



KUVIO 1. Sidosryhmäkaavio Alfamesta ja sen tärkeimmistä ulkoisista sidosryhmistä.

Työni kannalta olennaisimmat käytännön ohjeet saan projektin arkkitehdiltä, jolta kyselen jatkuvasti paljon tietoa Slack -ohjelmiston kautta ja videoneuvotteluissa. Käymme läpi esimerkiksi uuden projektin tiettyä osa-aluetta siten, että minä jaan arkkitehdille näyttöä. Hän opastaa minua avaamaan esimerkkipätkiä eri ohjelmista, auttaa selittämällä sieltä oleelliset ja välttämättömät asiat ja haastaa sitten minua päättämään asioita esimerkiksi vertaamalla uuden projektin toimeksiantoa vanhoissa projekteissa toteutettuihin asioihin. Jonkin verran apua ja neuvoja saan myös projektin muilta sovelluskehittäjiltä ja muilta kollegoilta. Projektin asiakasvastaavalta saan neuvoja esimerkiksi tuntien merkitsemiseen liittyen, ja hänen kanssaan keskustellaan myös päivittäin Scrum-palavereissa ja kahden viikon välein suunnittelupalavereissa tekemisen seurannan ja suunnittelun merkeissä.

2.3 Vuorovaikutustaidot työpaikalla

Tiimissä, jossa olen mukana, työtä tehdään eräänlaisella Scrum-tekniikalla. Tässä tapauksessa tämä tarkoittaa käytännössä sitä, että työt on jaettu kahden viikon sprintteihin ja pidämme päivittäin Scrum-palaverin Google Meetsin avulla. Palaverissa käymme aina läpi edellisen päivän aikana tehtyjä asioita ja mietimme seuraavaksi tehtäviä asioita. Lisäksi olemme jatkuvasti yhteydessä toisiimme Slack-sovelluksen kautta ja käytämme tarvittaessa Google Meets -sovellusta näytön jakamiseen kollegoille. Juttelemme paljon yleisistä työasioista ja muistakin asioista ainakin päivittäin kello kahden kahvipaussilla ja yleensä myös aamupäivän kahvitteletkinä ja lounaalla. Tarvittaessa pidämme sparrailulounaita ja lounasvalmennuksia erilaisiin tärkeisiin asioihin liittyen.

3 PÄIVÄKIRJARAPORTOINTI

Seurantaviikko 1

Maanantai

Tavoitteena oli saada kollegan kanssa valmiiksi yksi osa asiakasprojektista. Tehtävänä oli muokata esimerkki-JSON-tiedostoa Mule-vuon sisällä.

Aluksi oli hankaluuksia muistella, kuinka ylipäätään tiedostoja luetaan Java-ohjelmoinnissa, mutta lopulta löysin siihen hyvät neuvot. Tavallisella Google-haulla löysin lopulta keinoja siihen, kuinka JSON-tiedosto luetaan Javassa JSON-objektiksi (Karim, W. 2020. Converting JSONObject to HashMap) ja kuinka se edelleen muutetaan hashmap-tietorakenteeksi (Crunchify. 2020. How to Read JSON Object From File in Java?). Tätä tietoa hyödyntäen oli helppo muuttaa tietojen yhdistelyä siten, että tiettyjen ehtojen vallitessa JSON-tiedostoa muutetaan toisesta JSON-tiedostosta saaduilla arvoilla kovakoodattujen arvojen sijaan. Onnistuin lopulta tietoa hakemalla ja soveltamalla tänään tekemään sen, mitä toivoinkin saavani tehtyä.

Tiistai

Tiistain tavoitteena oli tehdä loppuun Java-luokka, jossa muutetaan JSON-tiedoston sisältämien avain-arvoparien arvoja ennalta määritettyjen ehtojen mukaan. Ehtojen tarkistuksen jälkeen Java-luokan metodit palauttavat arvoja, joita saadaan toisesta JSON-tiedostosta. Tämä pitäisi saada hyväksytyksi merge request -pyynnöllä projektin kehityshaaraan. Tämän jälkeen Scrumissa mietitään, mitä projektin osaa tutkin seuraavaksi.

Viimeksi toteuttamastani ratkaisusta löytyi puutteita. Yhden metodin osalta JSON-tiedosto puuttui kokonaan ja paluuarvot olivat edelleen kovakoodattuja. Täydensin toteutusta tältä osin valmiimmaksi ja varmistin yksikkötestillä, että homma toimii. Lähetin laatimani osan projektista sovellusarkkitehdin hyväksyttäväksi Gitin “git merge request” -toiminnolla. Opin siis tänään erityisesti JSON-tiedostojen yhdistämistä Javan avulla. Kovan työn jälkeen oli palkitsevaa nähdä toteutuksen toimivan yksikkötestissä.

Keskiviikko

Keskiviikkona odotin palautetta eilisestä merge requeststa ja kirjoittelen lopputyötä ja käyn koululla ruotsin tunnilla. Tämänhetkinen sprintti alkaa olla lopussa, joten tämänhetkisessä työprojektissa ei ole juuri nyt minulle paljonkaan tehtävää. Tutkin kuitenkin muiden tekemää koodia ja yritän ymmärtää sieltä yksityiskohtia, jotka eivät aiemmin ole auenneet.

Kollegoilla oli muita kiireitä, joten merge requestiani ei ehditty katselmoida. Löysin projektista yhden bugin, joka korjattiin. Ohjelma ei toiminut oikein, kun erään if-lauseen ehdon sisällä oli “||”, kun kyseisessä kohdassa oli tarkoitus olla “&&”. Ehdin myös tutkiskella hyviä keinoja organisaatiokaavion tekemiseen. Tänään opin, että myös kokeneet kollegat voivat tehdä virheitä, jotka myös minun on mahdollista havaita ja auttaa korjaamaan. Tälle päivälle asettamani tavoitteet ylittyivät sen osalta, että pääsin hyvin alkuun myös opinnäytetyön teossa ja organisaatiokaavion suunnittelussa. Haasteena oli erityisesti organisaatiokaavion asettelu, jossa en päässyt vielä hyvään tulokseen.

Torstai

Tänään lähdimme matkalle Helsinkiin, jossa pidetään lounassparraus nykyiseen projektiin liittyen ja seuraavan sprintin suunnittelutapaaminen, ja ennen kaikkea illalla osallistumme Great Place to Work 2020 -gaalaan. Viime vuonna Alfame sijoittui gaalassa toiseksi pienten yritysten sarjassa. Tänä vuonna tähtäin on keskisuurten yritysten sarjassa menestymisessä. Tätä tilannetta aion hyödyntää erityisesti sikäli, että pääsen tapaamaan Helsingissä työskenteleviä kollegoitani ja verkostoitumaan heidän kanssaan etätyöskentelyä paremmin.

Suunnittelupalaverissa päätettiin uuden projektin aloittamisesta ja nykyisen viimeistelystä. Päällimmäisenä jäi kuitenkin mieleen Great Place to Work -gaalan keskisuurten yritysten sarjan voitto. Tunnelma oli hieno. Onnistuin hyvin olemaan mukana tunnelmassa. Projektiin liittyen opin suunnittelupalaverissa hieman projektin suunnittelulogiikasta. Haasteellisinta oli ymmärtää, miksi tietyt ohjelman vaiheet tehdään juuri siinä järjestyksessä kuin on esitetty. Tärkein saavuttamani tavoite oli kuitenkin kollegoiden kanssa verkostoituminen, jossa pääsin hyvin alkuun.

Perjantai

Tänään onnistuin heräämään päivän Scrumiin kymmeneksi ja vähän mietittiin, että voisinko junassa paluumatkalla ainakin vähän katselmoida koodia. Muuten päivä meni lähinnä eilisistä juhlista toipumisen merkeissä. Tarkoitus on myös valmistautua mediahuomioon, jota gaalavoitosta eittämättä seuraa. Tässä voisi olla tilaisuus päästä itsekin lehden sivuille.

Sovimme, ketkä menevät toimistolle tapaamaan toimittajaa ja kuvaajaa. Itsekin ilmoittauduin vapaaehtoiseksi. Kuvaaja kävi toimistolla ottamassa meistä muutamia kuvia ja toimittaja kyseli kaikenlaista. Kerroimme parhaamme mukaan toiminnastamme. Onnistuin itsekin pääsemään paikallisen Keskipohjanmaa-lehden kuvaan, ja kertomani kokemus työhakemuksistani Alfamella oli otettu myös mukaan heidän juttuunsa, joten päivää voi pitää kaikin puolin onnistuneena.

Viikkoraportti

Opin viikon aikana ennen kaikkea siitä, kuinka Mule-flow voi keskustella tavallisen Java-luokan kanssa ja kuinka Javalla voidaan käsitellä Mule-flowlta saatua viestiä. Suurimpana haasteena oli ymmärtää, kuinka tieto kulkee Javan ja Mulen välillä. Tämä ei ollut välittömästi selvää pelkällä koodin tutkimisella. Vuodenvaihteessa keräämäni kokemus Java-ohjelmoinnista (Buchalka, T. Lochert, G. 2019. Java

Programming Masterclass for Software Developers) on kuitenkin ollut tässä suureksi avuksi ja antoi keinoja toimia joustavammin tiedon käsittelyssä ja muokkaamisessa tarpeiden mukaan.

Projektissa käytettävä Java sisältää kuitenkin minulle vieraita kirjastoja ja muun muassa Maven- ja Spring-tekniologioiden käyttöä, joihin en ole aiemmin tutustunut. Näistä pääsin hieman perille esimerkkien avulla, mutta joitain yksityiskohtia jäi vielä pimentoon. Olen oppinut käyttämään uusia kirjastoja, kuten Gson-kirjastoa, jolla on helppo muuttaa tekstiä JSON-muotoon ja takaisin. Olen myös oppinut erilaisia tapoja tiedostojen lukemiseen. Olen myös alkanut ymmärtää hieman Activiti-prosessikaavioiden toiminnasta, mutta siellä on vielä kuitenkin hieman epäselvyyksiä sen suhteen, miten Mule ja Activiti keskustelevat keskenään. Tästä selviämiseksi kyselin kysymyksiä projektin arkkitehdiltä ja sainkin niihin joitakin vastauksia vaikken kaikkea tietoa vielä täysin saanut sisäistettyä. Tämä vaatii lisää aikaa ja esimerkkejä.

Vaikeudet kehitysympäristön toimimaan saattamisessa edellisellä viikolla herättivät myös tällä viikolla ajatuksen siitä, että voisin olla mukana kehittämässä jonkinlaista sisäistä opasta tähän aiheeseen liittyen seuraavia uusia työntekijöitä ajatellen. Itse selvisin näistä vaikeuksista kyselemällä kollegoiden käyttämien työkalujen versionumeroita ja asentamalla mahdollisimman samat versiot kaikesta myös itselleni.

Seurantaviikko 2

Maanantai

Tänään oli tarkoitus oppia uuden projektin perustamiseen liittyviä asioita. Ensin ajetaan asiaan liittyvä skripti ja sitten opettelen, kuinka uusi projektihakemisto siirretään git:iin. Katsotaan lisäksi, mitä muuta tärkeää pitää asiaan liittyen oppia ja tietää.

Sainkin tehtyä paljon enemmän kuin luulin. Uusi projekti saatiin perustettua ja projektin arkkitehti antoi minulle tehtäväksi aloittaa ensimmäisen Mule-flown toteutuksen. Mule-flowssa haetaan tietokannasta tietoa monimutkaisella kyselyllä. Kysely on valmiiksi muotoiltu tiedostoon, josta se pitää ensin lukea ja sitten lähettää kysely palvelimelle. Tehtävä vaikutti aluksi mahdottomalta, mutta pikkuhiljaa aloin ymmärtämään ja löytämään osia ratkaisuun. Sain tämän melkein valmiiksi, HTTP-pyyntö ei vain jostain syystä toimi. Lähdän ensin selvittämään asiaa Mulesoftin dokumentaatiosta. (Mulesoft LLC. 2020.

HTTP Request Configuration Reference) Usein tämän tyyliin ongelmiin on jo aiemmin törmännyt joku muu, jolloin ratkaisu voi löytyä esimerkiksi asiaan liittyviltä nettisivuilta. (Stack Overflow. All questions.)

Tiistai

Sain aamulla viestin projektin arkkitehdiltä, että toteuttamani Mule-flow toimii hänellä. Hän oli vain vähän siistinyt koodia mutta muuten hän kehui, että hyvältä vaikuttaa tähän asti. Latasin Gitistä hänen tekemänsä muutokset, mutta minulla ohjelma kaatui edelleen kättelyvirheeseen palvelimen kanssa. Kyseessä lienee nyt jokin konekohtainen ongelma. Selvitämme asiaa myöhemmin tänään.

Ohjelma antoi virheilmoituksen “Received fatal alert: handshake_failure”. Tästä selvisi, että kyseessä oli sertifikaattiongelma. Ohjelma saatiin toimimaan muuttamalla http-pyynnön autentikointitapaa. Ongelma johtui siitä, että palvelin vaatii tietynlaisen OAuth-autentikoinnin ja siihen sopivan sertifikaatin. En vielä saanut syvällisempää ymmärrystä näiden autentikointitapojen ja sertifikaattien toiminnasta, joten projektin arkkitehdin tekemä päättely ja korjaustavan löytäminen jäivät minulle vielä hieman mysteeriksi. Asian tarkempi ymmärtäminen vaatisi OAuth-dokumentaation perehtymistä. (Campbell, B. 2019. OAuth 2.0 Mutual TLS Client Authentication and Certificate-Bound Access Tokens). Kuitenkin minulle annettujen esimerkkietojen avulla HTTP-request -konfiguraatioon (LIITE 12) tekemiäni korjausten jälkeen toteuttamani Mule-flow toimii ongelmitta ja palvelimelta saadaan haettua dataa.

Keskiviikko

Tänään on taas rikkonainen päivä. Aamusta jatkan Alfamella seuraavan Mule-flown tekemistä ja päivällä siirryn koululle ruotsin tunnille. Mahdollisuuksien mukaan palaan Alfamelle koulun jälkeen. Tarkoitus on testata projektin seuraavaa vaihetta, jossa palvelimelta saatua dataa jäsenellään osiin. Datassa on Arraylist Hashmappeja, jotka sisältävät ilmoittautumisia, joita pitäisi sitten ilmoittautumisen tyyppistä riippuen myöhemmässä vaiheessa yhdistää muutaman eri kategorian JSONeiksi.

Ennen puolta päivää ehdin toteuttaa lyhyen Mule-flown ja testin sille. Flowssa otetaan data vastaan Mule-viestinä ja lähetetään se tekemälleni uudelle Javaluokalle. Javaluokka ottaa taas datan vastaan ja siellä parsin sieltä datan seasta ilmoittautumiset yksitellen Hashmapiin. Myöhemmin nuo hashmapit on

tarkoitus lähettää yksi kerrallaan seuraavalle prosessille käsiteltäväksi. Tämä vaatii kuitenkin ensin Activiti-prosessikaavion piirtämistä. Onnistuin päivän luonteesta huolimatta tekemään enemmän kuin päivän alussa oletin. Haasteellisinta oli ymmärtää, kuinka tieto liikkuu Mule flowsta toiseen ja Mule ja Javan välillä. Tähän liittyen sain vihdoinkin eräänlaisen ahaa-elämyksen. Ymmärrän nyt sen tavan, miten Mule-flowiin tuodaan tietoa käyttämällä Springin required propertyä ja kuinka Mule-flowsta viedään dataa eteenpäin commit properties -toiminnolla (LIITE 2) tai Java luokassa Mule_invocation -toiminnolla. (LIITE 4). Tämä selvisi tutkimalla itse koodia ja varmistamalla asia Google Meets -keskustelussa projektin arkkitehdin kanssa. Tämän ymmärrettyäni minulla on huomattavasti paremmat edellytykset tehdä uusia Mule-floweja ymmärtäen, kuinka siellä otetaan vastaan ja lähetetään tietoa.

Torstai

Sain eilen tehtyä lopulta niin paljon, että tälle aamulle ei heti ollut selvää, mitä minun pitäisi seuraavaksi tehdä. Projektin arkkitehti työskentelee nyt vielä prosessikaavioiden ja prosessien käynnistämisen parissa. Kenties aamupäivän Scrumissa keksimme jonkin flown, jota voisin alkaa työstää. Mielenkiintoa olisi pohtia ainakin tietojen yhdistelemistä, mutta mahdollisesti se päätetäänkin tehdä Pythonilla Javan sijaan, jolloin minulla ei olisi osaamista siihen, enkä näe, että kokonaisen uuden ohjelmointikielen opettelu tässä vaiheessa olisi kovin järkevää, koska vielä on paljon muutakin opeteltavaa.

Päätettiin että aloitan tekemään sellaista Mule-flowia, jossa merkitään aiemmin haettu tieto käsitellyksi. Haasteena tässä oli minulle vieraan GraphQL-kyselykielen käyttö HTTP-requestin avulla. Tässä vaiheessa minun ei kuitenkaan tarvinnut lähteä opettelemaan itse GraphQL:ää vaan riitti, että opettelin lähettämään minulle valmiiksi annetun kyselyn HTTP-POST-menetelmällä. Tässä pystyin käyttämään hyödyksi aiemmin harjoittelemaani tiedostojen lukemista ja koin yksinkertaisimmaksi laittaa koko kyselyn tiedostoon, josta se sitten luetaan String-muuttujaan, joka lähetetään HTTP:llä palvelimelle. Lopulta tiedon lähetys palvelimelle onnistui projektin arkkitehdin avustuksella. Erityisesti sain apua siihen, kuinka selvittää tilanteesta, jossa palvelin ei hyväksy rivin vaihtoja kyselyssä, mutta tiedosto on kuitenkin sisällöltään pitkä ja sen tulisi olla tarvittaessa myös ihmisen luettavissa. Ratkaisuna tähän oli kehittää Java-luokka, joka muuttaa alun perin rivinvaihtoja sisältävän tiedoston rivinvaihdot palvelinta varten “/n” -merkinnöiksi.

Lisäksi suunniteltiin vielä erillistä Mule-flowia, jossa haetaan Dynamics-palvelimelta toteutustietoja. Myöhemmin on sitten tarkoitus yhdistellä näitä molemmilta palvelimilta saatuja tietoja uudelleenlaiseksi

tietopaketiiksi. Harjoittelin myös Activiti-prosessikaavion muokkaamista nykyiseen projektiin sopivaksi edellisen projektin esimerkin pohjalta. Pääsin tässä hyvin alkuun ja opin löytämään Activiti-kaaviosta ne kohdat, joissa oleellimmat tiedot esitellään. Oman oppimisen kannalta oleellisinta oli kuitenkin parantunut ymmärrys siitä, kuinka Javan avulla voidaan muokata Mulelta lähetettyä dataa.

Perjantai

Tänään on tarkoitus jatkaa eilen aloitettuja Mule-floweja ja tehdä niille stubit. Stubeja varten tarvitsee myös luoda JSON-tiedostoja, jotka sisältävät esimerkkidataa. Lisäksi ajateltiin kuitenkin, että voisin vähän opetella myös Pythonia, koska sillä on huomattavasti näppärämpi käsitellä JSON-muotoista tietoa kuin Javalla. Pythonia on kuitenkin aluksi tarkoitus käyttää lähinnä aiemmin haetun tiedon suodatukseen. Tätä opetellakseni pidän projektin arkkitehdin kanssa palaverin videoneuvottelun avulla. Päivän aikana opin tekemään yksinkertaisen Python skriptin (LIITE 3), jolla saadaan tehtyä filteri. Kyseisellä skriptillä haetaan JSONista toteutuskoodi, jolla voidaan myöhemmin etsiä tai suodattaa tietoa toisesta JSON:sta.

Haeskelin myös netistä tietoa Git-komennoista, kun oli tarve löytää komento, joka listaisi oman paikallisen haarani commitit, joita en ole vielä pushannut Gitlabiin. (Stack Overflow. 2020. Viewing unpushed git commits.)

Viikkoraportti

Tällä viikolla olen saanut lisää ymmärrystä projektin kokonaisuuteen varsinkin projektin alkupuolen osalta. Osaan nyt aloittaa uuden Mule 3 -projektin ja tehdä siihen tietynlaisia Mule-floweja. Olen myös oppinut tekemään stubeja, joiden avulla testivaiheessa voidaan käyttää esimerkkidataa. Tämän ansiosta yksikkötestejä tehtäessä ei enää tarvitse muodostella yhteyksiä palvelimille, vaan testejä voi ajaa vaikka junassa ilman jatkuvaa internet-yhteyttä.

Viikon paras hetki oli se, kun oivalsin, kuinka Javalla lähetetään tietoa tilanteessa, jossa halutaan käynnistää useita prosesseja, jotka ajavat samanmuotoista sisältöä käsitteleviä Mule-floweja rinnakkain (LIITE 4). Olen alkanut ymmärtää minulle aiemmin vieraalta vaikuttaneita tietorakenteita ja tärkeimpänä oppina olen alkanut ymmärtämään tiedon liikkumista Mulen sisällä sekä Mulen ja Javan sekä Mulen ja Activitin välillä. Hain usein lisätietoa muiden ohjelmoijien aiemmin tekemien kysymysten

vastauksista (Stack Overflow. 2020. Kotisivu). Usein sain myös apua suoraan kollegoilta Slack-chatissa. Viikko on ylittänyt odotukseni kaikin puolin Mulen käytön oppimisen suhteen. Nyt alan jo pikkuhiljaa uskoa, että tämän työn vaatimuksia vastaava tieto on mahdollista oppia tämän harjoittelujakson aikana. Tällä viikolla osaamiseni on kehittynyt suurin harppauksin pisteestä, jossa en vielä ymmärtänyt suurinta osaa Mulen perustoiminnoista pisteeseen, jossa voin periaatteessa ottaa vastaan tehtävän, jossa minua pyydetään toteuttamaan yksinkertainen Mule-flow ja saan sen tarvittaessa tehtyä suurimmalta osin itsenäisesti. Olen myös vaivihkaa aloittanut Pythonin opettelun, mutta siinä riittää vielä paljon opeteltavaa.

Seurantaviikko 3

Maanantai

Viikonlopun aikana kävin läpi Python kurssin (Udemy. 2019. Python for Absolute Beginners.) oppiakseni perussyntaksin projektissa käytettäviä Python-jäsentimiä varten. Kurssi oli nopea ja suoraviivainen ja koen että opin siitä tarvittavat perusasiat tai ainakin suurimman osan niistä. Tänään on tarkoitus soveltaa opittua käytäntöön ja tehdä Python-jäsennin, joka hakee JSON:sta tuotetunnisteen ja Mule-flow, joka käyttää kyseistä Python-skriptiä skriptauskomponentissa. Haasteena oli ensinnäkin selvittää tällaisen skriptauskomponentin tarkka syntaksi (LIITE 6). Kysyin aiheesta projektin arkkitehdiltä ja sain häneltä esimerkkipätkän Mule-flowsta, jossa tällaista komponenttia oli käytetty ja esimerkki-Python-skriptin. Muutaman tunnin näitä tutkittuani onnistuin mallia käyttäen ja soveltaen tekemään Python-skriptin, joka hakee JSON-tiedostosta tuotekoodin (LIITE 7). Tästä on jatkossa apua tietojen käsittelyssä. Python vaikuttaisi olevan hyvä työkalu tietojen käsittelyyn, ja sen opettelussa olen päässyt nyt hyvään alkuun.

Tiistai

Tänään työskentelen etänä kotoa käsin. Scrum-palaveriin osallistun normaalisti videoneuvottelun kautta. Tänään käydään läpi eilen tehty Mule-flow ja jatketaan aiemmin kesken jäänyttä Mule-flowia, joka merkitsee Concrete-palvelimelta saadun hakemuksen käsittelemiseksi, ja toista Mule-flowia, joka lähettää valmiiksi käsitellyn tiedon Dynamics-palvelimelle. Osoittautui että olin tehnyt hieman liian hankalasti

tuotteiden haun ja minun toteuttamani Mule-flow korvattiin suurimmaksi osaksi projektin arkkitehdin tekemällä yksinkertaisemmalla toteutuksella. (LIITE 8)

Tänään opin lisäksi muun muassa Prefer-tunnisteen käyttöä: HTTP-requestille viety tieto saadaan sen avulla takaisin Mule-flowiin logitusta varten. Opin projektin arkkitehdin kanssa keskustelemalla myös sen, että joka Mule-flowissa ei tarvitse lähettää "commit properties" -toiminnolla takaisin prosessiin niitä tietoja, jotka eivät ole Mule-flowin aikana muuttuneet: Kunhan tiedot vain on jossain vaiheessa viety prosessiin, niitä voidaan joka tapauksessa ottaa käsittelyyn missä tahansa prosessin myöhemmässä Mule-flowssa. Tämä auttaa minua välttämään turhaa tietojen pallottelua jatkossa.

Keskiviikko

Tämänkertainen sprintti alkaa taas olla päätöksessään ja tänään on luvassa uusi suunnittelupalaveri ja lisäksi lounasvalmennus ylläpitopalveluun liittyen. Ylläpitopalvelussa asiakas voi soittaa Alfamen Service centeriin ilmoittaakseen ongelmasta jossain Alfamen tarjoamassa integraatiossa tai muussa tuotteessa. On ollut puhetta siitä, että ylläpidon puolella ei ole tarpeeksi resurssia ja sitä pitäisi siis laajentaa isommalle porukalle. Minun osaltani tähän palveluun osallistuminen voi tulla kyseeseen muutaman kuukauden kuluttua, kun minulla kertyy vähän vielä lisää kokemusta.

Suunnittelupalaverissa päädyttiin lähinnä tuomaan esiin asioita, joita vielä pitää viimeistellä kahdessa viimeisimmässä projektissa. Minulle jäi toistaiseksi tehtäväksi Mule-flow, jolla lähetetään virhesähköposti, ja lisäksi minun pitää käydä läpi jo tehtyjä Mule-floweja ja lisätä sinne log-tulosteisiin kussakin Mule-flowssa käsittelyssä olevan hakemuksen Guid-tunniste. Sain tänään lisää tietoa Alfamen toiminnasta palvelujen ylläpidon osalta. Sain ymmärrystä siitä, kuinka kohdataan tilanne, jossa asiakkaalle aiemmin toteutetussa tuotteessa ilmenee myöhemmin ongelma, johon asiakas tarvitsee apua.

Torstai

Tänään aloitan log-muutokset lisäämällä Mule-floweihin uusia lokeja ja muokkaamalla vanhoja siten että kaikissa log-merkinnöissä esiintyy kulloinkin käsittelyssä olevan hakemuksen Guid-tunniste.

Tämän testaamiseksi myös niiden Mule-flowien testiluokkiin, joista Guid-tunniste puuttuu, täytyy lisätä esimerkki-Guid. Log-viesteissä toistetaan yleensä Mule-flown payloadin tulostusta Mule-flown eri

vaiheissa. Log-muutostehtävän lisäksi alan myös suorittaa virhesähköpostin lähetys-flowta. Testatessa käytetään SNMP-stubia, joka lähettää sähköpostin pelkästään suorituksessa olevan ohjelman komentoriville log-tulosteena.

Jäin hieman jumiin sähköpostin lähetyksen testaamiseen. Testaan virhepostin lähetystä antamalla prosessille tahallaan virheellisen hakemuksen käsiteltäväksi, jolloin näennäisesti prosessi etenee vaiheeseen, jossa virhesähköposti lähetetään, mutta jostain syystä viesti ei kuitenkaan tulostu ruudulle. Pyrin saamaan tähän apua kollegoilta mahdollisimman pian. Opin tässä kuitenkin periaatteet siitä, kuinka sähköpostin lähetystä testataan käyttäen näennäispalvelinta, joka lähettää sähköpostin vain ruudulle luettavaksi.

Perjantai

Tänään on tarkoitus saada virhesähköpostin lähetys toimintaan. Kollega antoi jo minulle neuvoja siitä, kuinka tässä käytetään Stub-SMTP-palvelinta testausvaiheessa. (LIITE 11). Olennaiseksi asiaksi osoittautui, että palvelimen täytyy odottaa viestien saapumista prosessista tarpeeksi kauan aikaa. Myös eräs projektin ulkopuolinen kollega auttoi minua ymmärtämään viestien vastaanottologiikkaa. Lopputuloksena pääsin tilanteeseen, jossa virhesähköposti tulee testissä toivotulla tavalla lokiin.

Tämän lisäksi sain toteutettua loppuun tehtävän, jossa minun piti lisätä olemassa oleviin Mule-floweihin log-tietoihin käsittelyssä olevan hakemuksen Guid. Tänään on vielä lounasvalmennus palkankorotusasioista, johon osallistun, vaikka asia ei suoranaisesti minua harjoittelijana kosketakaan.

Lisäksi iltapäivällä meillä on vielä viikoittainen kahvitteutilaisuus, jossa nostetaan esiin kuluneen viikon onnistumisia. Aion siinä kiittää minua auttaneita henkilöitä. Onnistuin tänään odotuksiani paremmin sähköpostin lähetyslogiikan selvittämisessä silloin kun sähköpostien lähettäminen on osana Mule-flowta.

Viikkoraportti

Tällä viikolla tuli taas opittua paljon uusia palasia siitä, mitä kaikkea Mule-flowien avulla voi tehdä, ennen kaikkea HTTP POST -pyynnön lähettämisestä, sähköpostien lähettämisestä, Mule-flowien

kommentoimisesta ja log-merkinnöistä. Kenties suurin ongelma oli se, kun sähköposti ei ilmestynyt odotetusti log-ruudulle silloin kun sitä yritin näyttää. Tässä auttoi se, kun minua opetettiin ymmärtämään, kuinka tällaisessa prosessissa viesti ei välity heti saman tien vaan sitä pitää erillisellä odotuskäskyllä odottaa sillä välin, kun prosessissa tapahtuu muun muassa tiedon hakua ja muokkausta.

Saamani palautteen perusteella aloin myös perehtyä Clean Code-kirjan (Martin, R. 2009. Clean Code: A handbook of agile software craftsmanship. 2009 Pearson Education, Inc.) oppeihin liittyen koodin asetteluun ja kommentointiin (luvut 4-5). Esimerkiksi en enää aio jättää kommentoituja koodirivejä koodiin, jonka vien Gittiin ja lisäksi kiinnitän huomiota koodin asetteluun niin, että se on mahdollisimman helppolukuista. Olen oppinut joka päivä uusia yksityiskohtia Mule-floweista ja niiden toiminnasta. Usein aamulla täysin pimennossa ollut asia tuli ratkaistuksi iltapäivään mennessä. Keskeistä oman oppimisen kannalta on ollut ymmärtää, kuinka tieto pidetään prosessissa tallessa ja kuinka sitä siirretään Mule-floweista Activiti-prosessille ja takaisin. Erityisen haasteellista tässä on ollut minulle aiemmin vieraan syntaksin ymmärtäminen ja niiden kohtien löytäminen, joissa tiedon siirtäminen varsinaisesti tapahtuu ja että näitä kohtia on monta erilaista riippuen siitä, mitä milloinkin ollaan tekemässä. Näiden asioiden tunnistaminen on kuitenkin auttanut minua kehittymään sillä tavalla, että seuraavan projektin kanssa osaisin jo huomattavasti sujuvammin luoda Mule-floweja, jotka vastaanottavat, käsittelevät ja lähettävät tietoa.

Seurantaviikko 4

Maanantai

Tänään keskityn ensisijaisesti tämän lopputyön hiomiseen, tarkistan selkeimpiä kirjoitusvirheitä ja epäloogisuuksia ja muokkaan sidosryhmäkaaviota selkeämmäksi. Projektin osalta tarkistetaan yksikkötesti, joka testaa ohjelman toimintaa mahdollisten käyttäjän kirjoittamien liian pitkien tekstien varalta. Esimerkiksi joillekin nimi- ja osoitekentille määritetään Pythonin avulla 17 – 250 merkkiä pitkät maksimipituudet, joiden ylittyessä ohjelman tulee osata katkaista teksti maksimimäärään merkkejä. Lisäksi tarkoitus on vielä tarkistaa log-merkinnät Mule-flowien virheenkäsitteilyosioissa, jotta ne ovat yhtenevät muiden osien kanssa.

Edellä mainittua yksikkötestiä luotaessa löytyi pari bugia, joista toisessa osaa tiedosta käsiteltiin päinvastaisilla ehdoilla kuin oli tarkoitus ja yhdessä kohtaa ohjelma salli liian pitkän työnantajan nimen

antamisen lomakkeella. Nämä korjattiin. Lisäksi työstin log-viestejä siten, että kaikissa luokissa kaikilla log-viesteillä on kategoria, joka määräytyy sen mukaan, missä Mule-flowssa viesti lähetetään, mukaan lukien Mule-flowien virheenkäsittelyosuudet. Tänään opin lisää Pythonista ja erityisesti sen, kuinka siinä voidaan käsitellä ja rajata merkkijonoa ja sen pituutta. Yhtenä haasteena oli ymmärtää Python-koodista se, millä tavalla Hashmap-tyyppistä dataa siellä käsitellään. Jälleen avuksi tuli projektin arkkitehti, joka antoi minulle muutaman esimerkkipätkän koodia ja kertoi, kuinka tieto siellä on jäsennelty. Python-osaamiseni ottikin tässä jälleen uuden askeleen eteenpäin.

Tiistai

Tänään meillä kerrotaan lounasvalmennuksessa uudesta Boomi-tekniikasta, joka olisi pilvipohjainen vaihtoehtoinen toteutustapa Mulen ja Activitin yhdistelmälle. Ymmärtääkseni esityksessä kerrotaan, mitä kyseisen tekniikan opettelu vaatisi ja kartoitetaan halukkuutta opetella ja osallistua kyseisellä tekniikalla toteutettaviin projekteihin tulevaisuudessa.

Esityksen jälkeen ryhdyin toimeen ja aloitin Boomi Essentials-kurssin (Boomi. 2020. Kotisivu.) läpikäymisen. Kesken tämän kuitenkin sain kaksi muuta tehtävää. Ensinnäkin minun piti avustaa uuden työntekijän työkalujen ja kehitysympäristön käyttöönotossa, ja toiseksikin sain asiakasprojektimme arkkitehdiltä bugiraportin.

Päivän aikana ehdin toteuttaa yhden muutoksen tuotteen tietojen käsittelyssä. Ohjelma antoi vääriä tietoja virheellisen tietojenyhdistelylogiikan vuoksi. Ongelma korjattiin asiakkaalta saadun palautteen mukaisesti. Tätä korjausta helpotti erityisesti aiemmin käyttöönotettu menetelmä, jossa tietojen yhdistelytapa oli määritelty erillisessä hakuehtoja ja niiden tuloksia sisältävässä JSON-tiedostossa. Ongelman korjaus vaati vain kolmen rivin muuttamista kyseisessä tiedostossa. Yksi ongelma jäi kuitenkin selvittämättä: Jostain syystä asiakkaalla ei ruudulle tulostunut lomakkeella mainittu kunnan nimi. Tätä selvitetään seuraavina päivinä.

Keskiviikko

Tämä päivä on rikkonainen palaverien ja koulunkäynnin vuoksi. Tarkoitus on kuitenkin jossain vaiheessa jatkaa eilisen bugin metsästystä. Aamupäivästä pidämme palaverin työpaikan psykologisesta turvallisuudesta. Tavoitteena on ymmärtääkseni ehkäistä ennalta tilanteet, jotka voisivat aiheuttaa

ristiriitoja henkilöiden välillä ja turvattomuuden tunnetta työpaikalla. Tavoitteena on varmistaa, että toimistolla on sellainen ilmapiiri, että vaikeistakin asioista ja esimerkiksi omista heikkouksista ja epävarmuuksista uskalletaan puhua. Tämä asia saatiin palaverissa alkuun. Todettiin asian tärkeys ja keskusteltiin asiasta alustavasti. Sovittiin että asian käsittely jatkuu tulevissa palavereissa ja että asiasta keskustellaan kaikkien kesken myös Slack-chatissa.

Edellä mainitun palaverin lisäksi pidimme myös lounaspalaverin huhtikuulle suunniteltuun Kroatian matkaan liittyen. Kävimme läpi matkajärjestelyitä ja suunnitelmia. Matkan toteutuminen on tosin epävarmaa koronaviruksen leviämisen vuoksi.

Virheen etsintään liittyen emme saaneet eilen selvittämättä jäänyttä virhettä toistumaan yksikkötestissä. Kun ongelma ei ole meidän testeissämme havaittavissa, ei sitä myöskään ole mahdollista tässä vaiheessa korjata. Projektin arkkitehti keskustelee aiheesta vielä asiakkaan kanssa ja selvittää, mistä tämä voisi johtua. Päivän tärkeimpänä antina pääsin pohdiskelemaan työilmapiirin ja ihmisten välisten kohtaamistilanteiden merkitystä työssä. Vaikka työssä ollaankin paljon tekemisissä koneiden kanssa, kaiken varsinaisen työn tekevät kuitenkin ihmiset, eikä tätä työtä voi tehdä yksin vaan työ vaatii onnistunutta kanssakäymistä kollegoiden kanssa.

Torstai

Tänään jatkan ainakin Boomi-kurssin läpikäyntiä. Mahdollisesti tänään tai lähipäivinä olisi tarkoitus myös tutustua projektin provisointiin eli eräänlaiseen asennukseen testipalvelimille.

Kollega havaitsi edellisen projektin tiedonkäsittelyssä ongelman, jonka johdosta yksi käsiteltävän lomakkeen kentistä jäi tyhjäksi. Ryhdyimme korjaamaan ongelmaa ja samalla päätimme myös muokata koko lomakkeen käsittelyä enemmän oliopohjaisen ohjelmoinnin suuntaan. Aiemmin lomakkeelta oli vain otettu arvot ylös yksittäisiin String-muuttujiin, mutta nyt teimme lomakkeesta olion, jonka arvoja sitten jatkokäsittelään.

Tärkein tästä oppimani asia oli, että kun käsitellään paljon tietoa, pitää olla erityisen tarkkana sen suhteen, että kaikki tarvittava tieto todellakin ylipäätään käsitellään jotenkin. Täytyy käydä isostakin tietomäärästä läpi kaikki kohdat ja tarkistaa, ettei mikään kenttä jää tyhjäksi. Tätä tarkistelua voidaan toki tehdä myös omin silmin logista, mutta silloin kun tietomäärä on erityisen massiivinen, voidaan tämä

tarkistus parhaiten tehdä yksikkötestillä, jossa oletetaan, että mikään käsittelyssä oleva kenttä ei ole arvoltaan tyhjä.

Loppupäivästä aikaa jäi Boomi Essentials -koulutukselle, ja sainkin sen käytyä läpi. Boomi vaikutti mielenkiintoiselta tavalta tehdä graafisesti samoja asioita, joita tähän asti on käsitelty tekstipohjaisessa editorissa.

Perjantai

Päivän alkaessa ei ole selvää, mitä tänään tekisin. Harkinnassa on Docker-kurssin aloittaminen. Pyysin kollegoilta suosituksia Docker-kurssin valinnassa. Valinta kohdistui Udemyn sivuilta löytyneeseen verkkokurssiin (Thompson, J. 2020. Docker for Java developers.). Tässä yhteydessä minulla oli aluksi hankaluuksia saada Gitistä kurssiin tarvittava materiaali. (Thompson, J. 2015. Springframeworkguru git repository). Löysin lopulta "git fetch origin" -komennon, jolla pääsin tarvittaviin git-haaroihin käsiksi (Git-scm. 2020. Download objects and refs from another repository.). Päivän loppuksi pääsin kurssissa osaan 3, johon mennessä Dockerin käsite on jo tullut selväksi ja jossa ajetaan erilaisia tietokantoja Docker-kontissa.

Viikkoraportti

Tämä viikko meni vähän kuin suvantovaiheessa. Varsinaista ohjelmointityötä oli vähemmän. Projektin arkkitehti on aloittanut kahden edellisen projektin provisioinnin. Minä en päässyt vielä tutustumaan projektin provisiointiin, mutta tarkoitus olisi, että seuraavan projektin yhteydessä pääsen ohjatusti aloittamaan myös sen puolen opettelua. Omalta osaltani olen valmistautunut siihen aiemmin suorittamalla Ansible-kurssilla ja nyt aloittamalla Dockerin opettelun.

Bugien korjaamiseen ja muuttuneisiin määrittelyksiin pääsin hieman tutustumaan alkuviikosta. Osittain sainkin korjattua projektia uusien määrittelyksien mukaiseksi mutta osittain asiakkaan kokema ongelma jäi projektin arkkitehdin ja asiakkaan välillä selvitettäväksi, koska ongelmaa ei saatu testeissä toistettua. Tältä osin kommunikointi asiakkaan kanssa on vielä jäänyt minulta vähän pimentoon tai arkkitehdin kautta suodatetun tiedon varaan. Mahdollisesti jossain vaiheessa myöhemmin pääsen myös itse suoraan jonkinlaiseen keskusteluun asiakkaan kanssa, mutta tässä vaiheessa minulla ei ole siihen vielä tarpeeksi kokemusta. Oman kehittymisen suurin eteneminen tapahtui ennen kaikkea Pythonin ymmärtämisessä, johon sain lisää opetusta projektin arkkitehdiltä.

Pääsin myös hyvään vauhtiin Boomin opettelussa ja alkuun Dockerin opettelussa. Dockerissa riittää vielä runsaasti opittavaa ennen kuin voin kuvitella olevani täysipainoisesti mukana projektien provisioinnissa.

Seurantaviikko 5

Maanantai

Tänään minun on tarkoitus jatkaa viime viikolla aloitettua Docker-kurssia. Muilta osin tiedossa ei ole mitään suoranaista työtehtävää. Mahdollisesti voidaan kuitenkin palata aiemmissa projekteissa havaittuihin virheisiin. Docker-kurssin jatkamisen lisäksi pääsin tänään tutustumaan projektin arkkitehdin esitykseen projektin provisiointiin liittyen. Pääpiirteissään projekti provisioidaan asiakkaan palvelimelle käyttäen Ansiblea ja Dockeria. Ansiblella rakennetaan Docker-kontit, joista sovellusta lopulta sitten suoritetaan. Asian tarkempi ymmärtäminen vaatii minulta kuitenkin vielä paljon aikaa ja asiaan liittyviin tiedostoihin tutustumista.

Docker-kurssissa etenin pisteeseen, jossa kolmasosa kurssista on suoritettu. Kurssin puitteissa olen oppinut rakentamaan Docker-tiedoston, joka luo konttiin Linux-imagena ja asentaa siihen Javan. Toivon saavani kurssin loppuun tämän viikon aikana. Käytännössä en vielä ymmärrä paljonkaan Dockerin toiminnasta mutta osaan esimerkkiä seuraten käyttää sitä tällaisiin yksinkertaisiin toteutuksiin.

Tiistai

Emme ole vielä saaneet sovittua seuraavan projektin yksityiskohtia, ja tekeminen on vähissä, joten keskityn tänään Docker-kurssin jatkamiseen ja tämän opinnäytetyön muotoiluun. Docker kurssia jatkan siitä, mihin eilen jäin, ja tavoitteena olisi päästä kurssissa puolen välin tienoille.

Tähän opinnäytetyöhön sain opinto-ohjaajaltani ohjeita siitä, mitkä osa-alueet tässä dokumentissa vielä vaativat kehittämistä. Käyn läpi tekstin tasaukset päivä päivältä ja tarkistan rivivälit. Käyn lisäksi dokumentin läpi kirjoitusvirheiden varalta ja lisään muutamia lähteitä, liitteitä ja määritelmiä. Pohdin myös tässä vaiheessa hieman tämän lopputyön tavoitteita. Työnantajan näkökulmasta lopputyön arvosanaa tärkeämpää on ilmeisesti ylipäättään valmistuminen ja insinöörin paperien saaminen asiaan liittyvää CV-merkintää varten. Henkilökohtaisesti toivon valmistuvani vähän paremmin kuin rimaa

hipoen, mutta täydellisyyden tavoittelun voin jättää kuitenkin tärkeysjärjestyksessä itse käytännön työssä oppimisen taakse.

Pääsin itse asiassa Docker-kurssissa tasan puoliväliin asti eli 52/104 videota on nyt katsottu. Tänään opin automatisoimaan Java-projektin Docker-imageksi pakkaamista projektin kääntämisen yhteydessä “mvn clean package docker:build” -komennolla ja suorittamaan sen “mvn docker:start” -komennolla. Sain myös lisää harjoitusta Dockerfile-tiedostojen tekemiseen.

Keskiviikko

Jouduin aamulla ilmoittautumaan sairaaksi, koska minulla alkoi herättyäni migreenikohtaus. Vaikka esimerkiksi pienessä flunssassa on hyvinkin mahdollista työskennellä etänä ja jatkaa töitä lähes normaaliin tapaan, on migreeni kuitenkin sellainen vaiva, joka estää tehokkaasti kaiken näyttöpäätteen äärellä työskentelyn. Töistä pois jäämiseen riitti ilmoitus Slackin kautta. Kollegat olivat onneksi ymmärtäväisiä asian suhteen ja toivottivat pikaista paranemista. Toivuin tästä kuitenkin siinä määrin että pääsin puolen päivän jälkeen koululle opinnäytetyö-workshoppiin. Workshopissa edistettiin etenkin lähdeviittausten ja pilkun käyttöä opinnäytetyössä. Opin myös refleктоimaan hieman paremmin aiemmin oppimaani ja lisäämään kuvausta oppimisestani.

Torstai

Tänään on toimistolla henkilöstöpäivä. Henkilöstöpäivä on Alfamella sellainen päivä, jolloin etätyö on kiellettyä ja kaikkien työntekijöiden tulee kerääntyä toimistolle työskentelemään. Lounasaikaan on tulossa henkilöstöinfo-tilaisuus. Tarkoitus on myös jatkaa tämän lopputyön kieliopin tarkistusta ja Docker-kurssia. Kurssissa on tarkoitus perehtyä tilanteeseen, jossa suoritetaan yhtä aikaa useita toistensa kanssa keskustelevia ohjelmia kukin omassa kontissaan.

Henkilöstöinfossa käsiteltiin taloudellista tilannetta, joka vaikutti olevan ihan hyvä, ja lisäksi muun muassa koronaviruksen vaikutusta työssäkäyntiin ja suunniteltuun Kroatian matkaan. Matkan toteutuminen vaikuttaa tässä vaiheessa epätodennäköiseltä, mutta kokonaan sitä ei vielä suljettu pois.

Docker-kurssissa jäin jumiin tilanteeseen, jossa kontin ajaminen komentoriviltä ei toimi, vaikka komento on tarkalleen kopioitu kurssin järjestäjän tekstitiedostosta. Ongelmana on ilmeisesti jokin muutos SSL-salauksessa. Lähetin kurssin järjestäjälle kysymyksen asiasta ja odotan vastausta siihen. En onnistunut tällä kertaa oikeastaan lainkaan oppimaan lisää asiasta, kun tekniset haasteet estivät etenemisen. Suurin osa päivästä meni tämän opinnäytetyön korjaamiseen ja tarkasteluun. Tarkastelun alla olivat tämän opinnäytetyön ohjaajan löytämät virheet ja puutteet. Työstä löytyi useita kirjoitusvirheitä, ja muutamia lähdeviitteitä ja käsitteitä oli tarpeen lisätä alun luetteloon.

Perjantai

Tänään on tarkoitus taas keskittyä Docker-kurssiin. Tavoitteena on, että tämän päivän jälkeen kaksi kolmasosaa kurssista olisi suoritettuna ja että saan kurssimateriaaleissa olevia esimerkkiohjelmaa korjattua ja ajettua ilman virheitä. Varsinaisissa työprojekteissa odotettavissa on, että ensi viikolla päästään aloittamaan seuraavaa projektia, joka vaikuttaisi alustavasti samankaltaiselta kuin aiemmat projektit. Pääpiirteissään tarkoituksena on hakea xml-muotoista tietoa yhdeltä palvelimelta, muuttaa se JSON-muotoon, käsitellä haettua tietoa halutulla tavalla ja lähettää tieto lopuksi toiselle palvelimelle.

Docker-kurssissa oli tällä kertaa vuorossa integraatiotestin suorittaminen. Kurssin projektissa suoritetaan komento `vmn verify`, jolla muodostetaan kontit `openjdk`:lle, tietokannoille ja web-palvelulle, suoritetaan testi ja tämän jälkeen suljetaan kontit. Ideana on demonstroida helposti toistettavissa olevaa testiä.

Sain Docker-kurssilla testin toimimaan. Haasteena oli se, että kurssimateriaali oli hieman vanhentunut ja esimerkkiprojektin toimimaan saaminen vaati monta pientä lisäystä ja korjausta. Onneksi kurssin yhteydessä on keskustelupalsta, josta löytyi ohjeita tilanteen korjaamiseen.

Lopulta sain projektin (Moilanen, P. 2020 Git. Spring-boot-docker-1 git repository.) toimimaan sen verran että testit saatiin suoritettua, mutta tässä vaiheessa vielä projektin varsinaisessa suorituksessa tulee virhe muodostettaessa yhteyttä tietokantaan. Ongelma lienee muuttuneissa vaatimuksissa SSL-salauksen suhteen. Pyrin selvittämään asiaa edelleen kysymällä asiasta kurssin keskustelupalstalla.

Viikkoraportti

Suurin tapahtuma kuluneella viikolla oli varmasti koronavirus-epidemian eteneminen, varsinkin aivan viikon lopussa. Tämä vaikuttaa työskentelyyni varsinkin siltä osin, että jatkossa töitä tehdään ensisijaisesti etänä, kunnes pahin paniikki on mennyt ohi. Asia mietityttää myös siltä kannalta, että tämä voi vaikuttaa asiakkaiden investointeihin ja sitä kautta resurssitarpeeseen Alfamella. Todella toivon, että tämä ei heikentäisi työllistymismahdollisuuksiani harjoittelujakson jälkeen, mutta nähtäväksi jää, kuinka tässä käy.

Tämän viikon aikana projektiryhmän tekeminen oli sangen vähissä. Töitä saatiin edellisillä viikolla tehtyä arvioitua nopeammin ja ainoastaan ryhmän arkkitehdille riitti tekemistä tälle viikolle. Onnistuin kuitenkin jatkamaan Docker-kurssia ja etenemään tämän opinnäytetyön kanssa. Harmillisesti jäin kuitenkin Docker-kurssissa hieman jumiin, kun kurssin harjoitusmateriaali ei ollut kaikilta osin ajan tasalla. Kenties tärkein opittu asia oli Wordin oikoluvun käyttö, jota en ollut aiemmin käyttänyt tai tulkinut oikein. Opinnäytetyön ohjaaja opasti kuitenkin minua siihen suuntaan, joten otin asiaksi perehtyä siihenkin aiheeseen. Opin sitä kautta tarkistamaan paremmin kirjoitusvirheitäni ja korjattua pilkkujen paikat oikeiksi.

Seurantaviikko 6

Maanantai

Tänään on suuret odotukset sen suhteen, että pääsisimme viimein aloittamaan uutta projektia. Mikäli näin käy, aletaan varmaankin projektin ensimmäinen sprintti ja projektin pilkkominen pienempiin osiin. Muussa tapauksessa jatkan verkkokurssien suorittamista.

Asiakasvastaava ei ollut vielä päässyt yhteisymmärrykseen asiakkaan kanssa projektin kustannuksista, mutta neuvottelu jatkuu vielä. Jatkoin Docker-kurssin loppuun saakka. Löysin lopulta kurssin Q&A-osiolta vastauksia ongelmiini. Tärkein edistymistä haitannut asia minulla oli ilmeisesti liian uusi versio mysql-tietokannasta, joka ei sitten ilmeisesti ollut yhteensopiva kurssin projektin kanssa. Vaihtamalla mysql versioon 5 alkoi kaikki lopulta toimia.

Olen nyt oppinut käyttämään Docker Swarmia siten, että voin tarvittaessa luoda kontteja ja jakaa niitä usealle eri palvelimelle kuorman tasaamiseksi ja virhesietoisuuden kasvattamiseksi. Kurssin

loppupuolella myös asennettiin yksinkertainen sovellus Digitalocean-pilvipalveluun. (Digitalocean. 2020. Kotisivu.) Opin tästä yhden mahdollisen tavan, miten valmiita sovelluksia voi asentaa palvelimille, joskin tuo kyseinen pilvipalvelu ei varmaankaan tule käyttöön. Omille projekteille se olisi liian kallis ja yrityksellä taas on muita vakiintuneita käytäntöjä.

Tiistai

Tänään saimme raportin vielä yhdestä bugista edellisessä projektissa. Ainakin yksi kenttä tuotteen tietojen yhdistelyyn liittyen on kirjoitettu väärin ja Dynamics palvelin ei tietenkään anna tallentaa vääränlaista tietoa. Ongelma korjataan sillä tavalla, että etsitään koko projektista kyseinen väärä merkkijono ja korvataan se oikealla, muokataan yksikkötestit vastaamaan uutta tilannetta ja käännetään projekti Mavenilla uudestaan. Lopuksi vielä sovellus tulisi asentaa uudestaan asiakkaan testiympäristöön käyttäen Ansiblea ja Dockeria. Korjaus saatiin tehtyä ja käännettyä ja vastaavat yksikkötestit korjattua. Huomenna on vuorossa korjauksen asennus. Tässä pääsin harjoittelemaan toisen kehittäjän tekemän virheen korjaamista. Tällaisen toimeksiannon yhteydessä ei oteta kantaa siihen, kuka virheen on tehnyt, vaan keskitytään sen korjaamiseen.

Keskiviikko

Projektin uudelleenasettamiseen oli olemassa valmiit ohjeet, mutta ohjeita seurattaessa sain kuitenkin virheilmoituksen Ansiblen .yml-tiedoston virheellisyydestä. Kysyin asiasta projektin arkkitehdiltä ja osoittautui, että minun pitikin vielä ensiksi kääntää erikseen tiettyjä vaadittavia kirjastoja, joita on hiljattain päivitetty. Projektin asentaminen asiakkaan testiympäristöön vaatii myös ssh-avaimen asentamisen asiakkaan ympäristöön. Opin tänään ainakin valmiiksi tehdyn Ansible-playbookin suorittamisen arkkitehdiltä saatujen ohjeiden mukaan. Playbookissa tehdään ohjelman ja sen vaatimien taustasovellusten paketointi Docker-konteiksi ja niiden käyttöönotto palvelimella. Tulevaisuudessa pitäisi vielä oppia tekemään Ansible-playbook -kokoelmat tuleville projekteille. Oman oppimisen kannalta tämä oli vielä hieman sekavaa ja vaatii vielä lisäperehtymistä sekä Ansibleen että Dockeriin, jotta hahmotan, mitä Ansible-playbookissa oikein tarkalleen ottaen tapahtuu. Varsinkin kun jouduin työskentelemään kotona ja valvomaan samalla koronavirus-pandemian vuoksi kotona etäopiskelevia lapsia, on työrauha välillä ollut katkolla ja sitä myöten keskittyminen pääsee välillä herpaantumaan. Tässä kohtaa kollegat kuitenkin ovat olleet ymmärtäväisiä ja tarvittaessa selostaneet saman asian toisenkin kerran.

Yleisesti ottaen tällaisessa projektin provisioinnissa kyse on kuitenkin siitä, että rakennetaan Docker-kontit, siirretään ne asiakkaan ympäristöön ja otetaan ne käyttöön kahdella Ansible-playbookilla siten, että tämä kaikki tapahtuu aina ennustettavasti ja samassa järjestyksessä.

Torstai

Tänään aloitetaan uusi projekti jakamalla projektin eri vaiheet pienemmiksi tehtäviksi projektin backlogille, josta ne sitten siirretään uuteen sprinttiin järjestyksessä. Tähän saadaan vinkkiä projektin arkkitehdin aiemmin luomasta määrittelydokumentista. Projektissa on tarkoitus ensin hakea XML-tiedosto palvelimelta, kerätä tieto XML:n sisältä, jäsenellä tieto toiselta palvelimelta haettuun Dynamics-pohjaan ja lähettää tieto lopulta Dynamics- ja Contentful-palvelimille. Projektin pilkkominen osiin toteutettiin lopulta siten, että minä loin uusia tehtäviä projektin tehtävälstaan eli backlogiin kollegan neuvojen perusteella. Minä jaoin näyttöä videoneuvottelussa, kyselin ja sain neuvoja kokeneemalta kollegalta. Projekti jaettiin vaiheisiin, joista yksi oli projektin prosessikaavion piirto, kolme eri vaihetta projektin provisioinnille ja omat vaiheensa jokaiselle Mule-flowlle ja sen yksikkötestille. Tästä sain projektinhallinnallisen osaamisen harjoitusta ja tästä on paljon hyötyä, jos aion joskus edistyä kehittäjästä arkkitehdiksi.

Projektin jaottelun jälkeen aloitettiin prosessikaavion piirtäminen. Myös tässä sain samaan tapaan neuvoja kollegalta videoneuvottelussa. Tällä tavalla käytännön tekemisellä saa nopeasti hyvän rutiinin projektin alkuvaiheisiin ja saa saman tien kysytyä itseä askarruttavia käytännön asioita, kuten missä kohtaa Activitissa määritellään Mulen puolelle tarvittavat asiat.

Perjantai

Tänään minun olisi tarkoitus aloittaa uusi Mule-flow, jossa tarkastellaan XML-tiedoston kelpoisuutta. Jotta XML-tiedosto läpäisee tarkistuksen, siitä tulee löytyä Koulutustieto, joka sisältää "Alkupvm", "Hinnoittelu" ja "Kustannuspaikka" -kentät.

Aloitin käyttämällä Mulen Choice-elementtiä, jolla Mule-flown voi ohjata haarautumaan eri vaihtoehtoihin tiettyjen ehtojen mukaan. Tällä tavalla toteutettuna Mule-flowsta tuli kuitenkin turhan sekavan oloinen ja huonosti luettava. Seuraavaksi yritin saada Mule-flowsta yksinkertaisempaa

käyttämällä Mulen Validator-elementtiä. Tämä aiheutti kuitenkin häiriöilmoituksen Mulen Extensionmanagerin käyttöönottoon liittyen. Tätä en saanut ratkaistua, joten asian selvitys siirtyy seuraavaan viikkoon.

Viikkoraportti

Mielestäni mielenkiintoisin ja tärkein tällä viikolla opittu asia oli se, kuinka projekti aluksi paloitellaan pienemmiksi kokonaisuuksiksi, joita on mahdollista tehdä toisistaan riippumatta ja lyhyessä ajassa. Samalla olen oppinut hallinnoimaan projektin backlogia tarvittaessa. Pääsin tekemään käytännössä myös Activiti-prosessikaaviota juuri nimettyjen työtehtävien mukaan. Sain myös alustavaa opastusta projektin asentamiseen liittyen, vaikka en siitä puolesta vielä kaikkea tarvittavaa tietoa onnistunutkaan sisäistämään. Työmotivaation kannalta olennaisinta oli kuitenkin, että päästiin vihdoinkin verkkokurssien sijaan tekemään oikeaa uutta projektia. Odotukset ovatkin korkealla, että seuraavan muutaman viikon aikana pääsen oppimaan taas paljon enemmän varsinaista ohjelmointia.

Seurantaviikko 7

Maanantai

En osannut tulkita viime viikolla saamaani virheilmoitusta, joten lähdin kysymään apua kollegoilta. Projektin arkkitehti ehdotti, että tekisin oman XML-tarkistajan Javalla. Tämä oli mielestäni hyvä idea. Tarkoituksena oli tehdä sellainen Java-luokka, joka tarkistaa, onko XML:ssä vaaditut elementit. Mikäli kaikki tarpeellinen on tiedossa mukana, jatketaan prosessia. Mikäli taas jokin tieto puuttuu, annetaan virheilmoitus ja lopetetaan kyseisen XML:n käsittely. Java-luokkaa varten XML-tiedosto muunnetaan ensin MAP-muotoiseksi Mule-flowssa.

Ensimmäinen haaste oli selvittää, miten ylipäätään lähetetään virheviesti Java-luokasta Mule-flowlle. Hain tietoa ensin Google-hakukoneen avulla ja löysinkin esimerkkejä TransformerExceptionistä (Codota. 2020. Best Javacode snippets using org.mule.runtime.core.api.transformer.TransformerException) ja siitä kuinka virheviesti lähetetään (Stack Overflow. 2020. How to throw a custom exception from inside Java Transformer in Mule ESB 3.8?).

Kun tämä oli saatu selville, pääsin koodaamaan Javalla muutaman if-ehdon avulla toimivan XML-tarkistajan. Tärkein oppimani asia tässä oli ilman muuta tuo virheviestien käsittely ja siirtäminen Javasta Mule-flowlle. Perjantain ongelmien jälkeen koen onnistuneeni tässä paremmin kuin odotin.

Tiistai

Päivän aluksi ei tällä kertaa ollut välittömästi selvää, mitä lähtisin tekemään. Kysyin asiasta projektin arkkitehdiltä, ja hän neuvoi vain katsomaan tehtävälustauksesta seuraavan tehtävän, jota kukaan ei vielä ole aloittanut. Valitsin sieltä datan jäsentelytehtävän, jossa otetaan tietoja sisään Mule-flowiin kolmesta eri JSON:sta ja luodaan niistä yksi uusi JSON asiakkaalta saadun määritelmän mukaisesti. Ensimmäinen haaste, oli etten aluksi saanut edes Mule-flown pohjaa toimimaan ilman mitään tuotuja tietoja. Tunnin päähkäilyn jälkeen havaitsin, että xml-tiedoston nimi oli kirjoitettu väärin yhdessä kohtaa yksikkötestiluokassa.

Kun lopulta pääsin alkuun, oli seuraavana haasteena muistella taas Pythonin käyttöä ja opetella sitä lisää. Katsoin ensin mallia aiemmin tehdyistä Python-skripteistä ja pääsin sen perusteella etenemään. Lisäksi pidimme vielä videoneuvottelun, jossa projektin arkkitehti opasti minua muun muassa sen suhteen, kuinka Pythonissa voidaan käyttää Dict-rakennetta korvaamaan suuri määrä peräkkäisiä if-lauseita (Liite 11). Jatkoin skriptin tekoa esimerkkietoa soveltaen. Jäin lopulta jumiin vaiheeseen, jossa skriptiltä saatua JSON-tiedostoa olisi pitänyt kokeilla lähettää palvelimelle. Tiedon lähettämistä voi testata helpoiten REST-asiakassovelluksella, kuten esimerkiksi Insomnia (Insomnia. 2020. Insomnia REST client.). Tässä tapauksessa kuitenkin palvelin ei hyväksynyt yhtä kenttää JSONista. Tämän selvitystä jatketaan huomenna.

Oman oppimiseni kannalta olennaista tässä oli saada hieman rutiinia tiedon jäsentelyyn Pythonin avulla. Olen yllätynyt siitä, kuinka nopeasti alan oppia taas uuden ohjelmointikielen käyttöä. Pienen alkuhaparoinnin jälkeen Pythonin käyttö alkaa vaikuttaa todella kätevältä. Projektin arkkitehdin opastus on tässä ollut todella hyödyllistä. Olen voinut häneltä varmistaa ja kysellä epäselviksi jääneitä asioita ja saanut yksityiskohtaisia neuvoja Python-syntaksin opetteluun.

Keskiviikko

Tänään olisi tarkoitus jatkaa eilen aloitettua Mule-flowta, jossa jäsennellään useasta eri JSON-tiedostosta saapuvaa tietoa yhteen uuteen JSON-tiedostoon. Sitten kun se on saatu kuntoon, tulisi vielä toteuttaa toinen Mule-flow, jossa kerätty tieto lähetetään palvelimelle. Projektin arkkitehti oli lähettänyt asiakkaalle kysymyksen liittyen eilen havaittuun tietokenttään, jota palvelin ei hyväksynyt. Selvisi, että muista tietokentistä poiketen juuri kyseisessä kentän nimessä esiintyi yksi iso kirjain. Tämän tiedon kanssa varustautuneena uusi kokeilu tiedon lähettämiseksi Insomnian avulla onnistui. Tämän jälkeen alkoi vastaavan tiedonlähetys-toiminnon toteutus Mulen avulla. Sain tässä hyvää kertausta siitä, kuinka rakennetaan Stub, joka korvaa yksikkötestissä oikean palvelimen ja kuinka rakennetaan http-request, jossa käytetään body-elementissä Mule flowiin prosessista tuotavaa JSON-tietoa. Lisäksi sain oppitunnin siitä, että joskus kovalla työllä toteutettu ratkaisu voidaan korvata toisella ratkaisulla heti kun parempi tai yksinkertaisempi ratkaisu keksitään. Viime perjantainen ongelma Mulen validator-komponentin kanssa nimittäin ratkesi sillä, että yksikkötestiluokkaan kirjoitettiin addBuilders-metodi (LIITE 14), jolla extensionManager otetaan käyttöön.

Torstai

Tänään jatkan projektin arkkitehdin aiemmin aloittamaa Python-parseria, jossa XML-tiedostosta otetaan talteen tarvittavia tietoja ja koostetaan niistä JSON-muotoinen dokumentti. Parserissa oli valmiiksi tehtynä yhdelle palvelimelle tarvittavien tietojen osuus, nyt sinne pitää vielä lisätä toiselle palvelimelle tarvittavien tietojen osuus. Haasteena tässä on hieman aiempaa monimutkaisempi logiikka, jossa muun muassa pitää tarkistaa, esiintyykö tietolähteessä opiskelumuo, joka on "Verkko-opiskelu". Lähdetiedossa saattaa lisäksi olla tietyn kurssin opiskelumuoja joko listana tai yksittäisinä kappaleina, jolloin niitä pitää niiden määrästä riippuen käsitellä eri tavoin. Tässä oli minulle hieman lisää Pythonin opettelu tarvetta.

Haasteena tässä oli se, että Python päättää itse olion tyyppin, enkä aluksi täysin ymmärtänyt, kuinka toimitaan tilanteessa, jossa tieto saattaa olla lista tai yksittäinen tieto ja vain listaan voidaan käyttää get-metodia. Vaarana oli, että yritetään käyttää get metodia yksittäiseen tietoon, jolle kyseistä metodia ei ole olemassa ja tämä kaataisi koko parserin.

Jouduin muun muassa hakemaan tietoa siitä, kuinka Pythonissa tarkistetaan olion tyyppi. (W3Schools. 2020. Python isinstance() Function.). Tämän selvitettyäni pystyin ratkaisemaan asian tekemällä if-lauseen, jossa tarkistetaan olion tyyppi ja haarauttamalla koodin sen mukaan. (LIITE 9). Tämä oli sangen

mielenkiintoinen pikku aivopähkinä, ja opin sen avulla taas lisää aavistuksen monimutkaisempaa Pythonin käyttöä.

Perjantai

Sain aamulla migreenikohtauksen ja työnteko jäi vähäisemmäksi. Yritin aluksi jatkaa eilistä työtä ja sainkin juuri ja juuri hiottua parserin loppuun ja testattua sen toimintaa, ennen kuin jouduin keskeyttämään työskentelyn huonon voinnin vuoksi. Kun iltapäivällä palasin työnteon äärelle, sain tietää projektin arkkitehdiltä, että hän oli jakanut työstämäni parserin kahteen osaan. Ryhdyin testaamaan sitä ja löysin ohjelmasta virheen. Tähän asti olimme olettaneet, että Päiväkurssi-muotoisia opintoja on dokumentissa aina joko yksi tai useampia. Löysin kuitenkin esimerkin, jossa näitä ei ollut dokumentissa lainkaan ja tällaisen dokumentin kohdalla ohjelma kaatui siihen, että “Nonetype”-tyyppiselle oliolle yritettiin käyttää get-metodia. Lisähaasteena Pythonin isinstance-metodi nähtävästi ei tunne “NoneType” -tyyppiä, joten sillä metodilla ei voida tarkistaa, onko olio “NoneType”. Tähän ratkaisuksi keksin sitten tarkistaa tavallaan toisin päin, että onko olio jotain tunnettua muotoa ja jos ei ole niin sitten käsitellään dokumentti eri tavalla kuin muussa tapauksessa. Tästä sainkin arvokasta tietoa siitä, kuinka tiedonkäsittelyssä Pythonilla voidaan tarkistaa, onko kyseinen tieto ylipäätään olemassa vai ei. Vastaavia tarkistuksia täytynee tehdä jatkossa myös ennakoivasti.

Viikkoraportti

Viikon teemana tällä kertaa olivat erilaiset virheidenkäsittelytavat, virhetilanteet ja tarkistukset niiden varalle. Opin välttämään tekemästä oletuksia lähdemateriaalin sisällön ja rakenteen suhteen silloin, kun lähdemateriaali voi olla hyvin vaihtelevaa ja sen rakennetta ei ole määritelty muuten kuin muutaman esimerkin avulla. Sain runsaasti hyvää harjoitusta myös tiedon jäsentelyyn ja tietorakenteen tutkimiseen loggaamalla ohjelman eri vaiheita. Yhtenä haasteena Pythonin debuggaus Intellijn avulla ei oikein tuntunut toimivan olettamallani tavalla. Debuggeri toimii vain testi-javaluokan sisällä ja hyppää yhdellä stepillä suoraan koko Mule-flown ja sen sisältämän Python-parserin yli. Pythonin opettelussa käytin sen vuoksi runsaasti väliaikaisia print()-lausekkeita selvittääkseni, mitä tietoa ajon aikana missäkin muuttujassa on. Vaikka tällä tavalla sain kuitenkin kaiken tarvittavan tiedon, vaati ongelmien löytäminen usein monta peräkkäistä ohjelman kääntämistä ja ajamista pienin muutoksin. Sitkeällä työllä onnistuin

kuitenkin lopulta saamaan haluamani asiat aikaiseksi ja odotan seuraavalta viikolta uusia haasteita, kuten esimerkiksi ohjelman muuttujainventaarioiden laatimiseen tutustumista.

Seurantaviikko 8

Maanantai

Tänään olisi tarkoitus toteuttaa Mule-flow, jossa haetaan Dynamics-palvelimelta yhteys henkilön tietoja ja haetaan niiden tietojen perusteella tietoja Contentful CMS -palvelimelta. Haku Dynamics CRM-palvelimelta toteutetaan HTTP-pyyntönä. HTTP-pyyntöä varten luodaan ensin hakufiltteri, jossa käytetään tietoja, joita haetaan aiemmin haetun xml:n tiedoista, jotka on jo valmiiksi yhdistetty JSON-tietorakenteeseen. Tämän toteuttamisessa pystyin käyttämään pienin muokkauksin avuksi projektin arkkitehdin luomaa Mule-flowta, jossa oli jo toteutettu vastaavanlainen toiminto. Opin tässä uudenlaisen tavan myös tiedonhaun Stub-testiliitynnän toteutukseen, jossa käytetään Mule-flowia, joka palauttaa vakiosisältöisen HTTP-vastauksen näköisen tiedon. Tämän vastauksen luomiseksi piti ensin tehdä yksi oikea haku asiakkaan testipalvelimelta. Tähän oikeaan hakuun käytin taas aluksi Insomnia-sovellusta.

Tiistai

Tänään on tarkoitus aloittaa harjoittelemaan projektin provisiointia. Provisiointiin kuuluu inventaarioiden rakentaminen ja Ansible-playbookien laatiminen. Inventaarioissa luetellaan ympäristömuuttujat, kuten esimerkiksi palvelinten osoitteet, portit ja muut asetustiedot sekä joitakin oletusarvoja tietyille muille muuttujille. Ansible-playbookissa taas on viittaukset ohjelman application.properties -arvojen ja ympäristömuuttujien välillä ja lisäksi siellä määritetään Docker-kontit, joissa ohjelmaa ajetaan. Playbookit määritellään erikseen paikallisesti ajettaviksi testaamista varten ja toisessa Ansible-playbookissa on versio samoista asioista asiakkaan testipalvelimella ja tuotantopalvelimella ajoa varten.

Käytännössä helpon tämän prosessin pystyi oppimaan kopioimalla ja muokkaamalla inventaarioita ja Ansible-playbookeja, joita on käytetty edellisissä projekteissa. Esimerkiksi tiettyjen palvelimien tietoja löytyi jo valmiiksi edellisestä projektista. Sieltä täytyi tosin poistaa sellaisen palvelimen tietoja, jota tässä projektissa ei ole käytössä. Loput muuttujat täytyi tarkistaa ohjelman application properties -tiedoista soveltamalla aiemmin käytössä olleen kaltaisia nimeämiskäytäntöjä. Käytännössä ohjelman

omat `application.properties`in kovakoodatut arvot muutettiin ympäristömuuttujiksi ja tarvittavat arvot sieltä siirrettiin sitten inventaarioihin. Lopuksi käytiin vielä projektin arkkitehdin kanssa läpi oikeat komennot, joilla `playbook`it suoritetaan. Opin tässä ymmärtämään paremmin sitä, kuinka ohjelma otetaan käyttöön. Tätä ennen oikeastaan kaikissa ohjelmointitöissä hiljattain suorittamaani Docker-kurssia lukuun ottamatta olen ajanut koodaamaani sovellusta ainoastaan sillä samalla koneella, jolla olen sovelluksen myös koodannut. Nyt sen sijaan valmistaudutaan viemään sovellus Docker-konttina asiakkaan testiympäristöön.

En vielä saanut provisiointia toimimaan enkä täysin ymmärrä, miksi se ei toimi tai mitä vielä pitää tehdä, jotta se saadaan toimimaan. Ehkä tämä asia kuitenkin alkaa ajan myötä hahmottumaan, kuten monien muidenkin asioiden kanssa tässä työssä on tapahtunut.

Keskiviikko

Tänään on tarkoitus tutustua aiemmin toteutettuun projektiin, jossa on käsitelty Contentful-palvelimen tietoja. Tarkoitus olisi soveltaa samaa Contentful-rajapintaa tiedonkäsittelyyn myös nykyisessä projektissa. Ensimmäinen haaste on hakea Contentful-palvelimelta tietue sellaisen syöttötiedon perusteella, joka on aiemmin haettu Dynamics-palvelimelta. Tämän avulla olisi sitten tarkoitus jatkaa tiedon yhdistelyä JSON-tiedostoon, jossa on tietoja Contentful-toteutuksesta.

Tässä kohtaa törmäsin isoihin haasteisiin. Aluksi en oikein edes ymmärtänyt Contentful-rajapinnan toimintaa, miten sieltä edes pitäisi saada tietoja kaivettua. Sitten kun lopulta pitkän pätkäilyn perusteella tunnistin sieltä metodin, jolla tietoa voi hakea esimerkiksi kunnan nimen perusteella, en kuitenkaan onnistunut saamaan vastaukseksi muuta kuin tyhjää tai virheilmoituksia. Lopulta lähdin kääntämään myös aiempaa projektia kokeillakseni sieltä löytyviä testiluokkia. Kun nämäkään eivät toimineet, päädyin siihen tulokseen, että vika ei välttämättä olekaan omassa koodissani vaan kenties rajapinnassa on nyt joku ongelma. Tätä on tarkoitus selvittää huomenna projektin arkkitehdin kanssa.

Torstai

Tänään on sikäli suuri päivä, että minulla on tänään haastattelu, jonka pohjalta päätetään työskentelyni jatkosta toukokuun loppuun päättyvän harjoittelujakson jälkeen. Lisäksi tarkoitus on jatkaa eilistä ongelmanratkaisua Contentful-rajapinnan kanssa ja lisäksi jatkan myös projektin provisiointia

muuttamalla projektin kovakoodattuja `app.properties`-muuttujia ympäristömuuttujiksi. Lisäksi on tarkoitus myös selvittää, mitkä kaikki arvot täytyy jollain tavalla salata.

Tein yksikkötestin, joka testaa Contentful-rajapinnan toimintaa, ja pyysin projektin arkkitehtiä tutkimaan rajapinnan toimivuutta sen avulla. Pian tämän jälkeen projektin arkkitehti raportoi, että hänelläkin on haasteita saada muuta kuin 404-virhe ilmoitusta, joka tarkoittaa, että tietoja ei palvelimelta löydy tai saatu. Tämän selvittäminen jatkuu sillä, että asiaa tarkistetaan asiakkaan kautta. Jatkoisin myös provisioinnin muuttujaluetteloiden täyttämistä. Tässä vaiheessa salattaviksi tarkoitettuja tietojen arvoja ei viedä muuttujaluetteloon, vaan niitä vastaavien avainten perään kirjoitetaan "Salataan". Ensi viikolla on tarkoitus käydä läpi, minkälaisista salauksista käytetään. Mahdollisesti kuitenkin Ansiblen vault voisi tulla käyttöön. Opin tässä eniten sitä rutiinia, että käydään Mule-floweista läpi kovakoodattuja muuttujien arvoja, muutetaan ne viittauksiksi `app.properties`-tiedostoon, jossa taas viitataan Ansible-playbookiin, josta taas viitataan `group-vars`-kansiossa oleviin tiedostoihin, joissa ympäristömuuttujat on lopulta määritetty. Opin myös, miksi näin tehdään: Tällä tavoin toteutettuna ohjelmaa ei tarvitse kääntää kokonaan uudestaan joka kerta, kun jonkun muuttujan arvoa halutaan muuttaa.

Haastattelu sujui mielestäni hyvin. Allekirjoitimme työsopimuksen alkamaan harjoittelujakson päätyttyä.

Perjantai

Tänään on yhden tiimissäni työskennelleen kollegan viimeinen päivä Alfamella. Hän siirtyy tämän jälkeen toisen yrityksen palvelukseen. Tänään käydään myös palaverissa läpi muutenkin muutoksia tämän ja muutaman muun projektin tiimeihin. Toisessa palaverissa taas esitellään erään ainakin minulle aiemmin tuntemattoman asiakkuuden projekteja. Tämän lisäksi tarkoitus olisi, että jatkan provisiointihommia.

Provisiointipuolella inventaariot on jaettu kolmeen osaan. Local-puolella on muuttujat, joita käytetään silloin, kun projektin kontteja testataan omalla tietokoneella. Test-puolen muuttujia käytetään, kun kontit asennetaan asiakkaan testiympäristöön. Prod-puolelle taas tulevat varsinaisen tuotantoympäristön muuttujat. Aiemmin olen perehtynyt local-inventaariotiedostoon, nyt on test-tiedoston vuoro. Sain

päivän aikana local- ja test-inventaariot siihen malliin, että niihin tarvitsee enää lisätä salasanojen salaus. Edelleen käytäntö on, että tutkitaan ohjelmasta käytettyjä muuttujia, jotka voisi olla hyvä saada muutettavaksi ilman ohjelman uudelleenkäyntöä, lisätään niistä viittaus projektin app.propertiesiin, josta viitataan Ansible-playbookiin, josta viitataan group-vars-kansiossa määritettäviin ympäristömuuttujiin. Tässä vaiheessa tarkistetaan myös tarvittavat muutokset muuttujien arvoihin silloin kun ohjelmistoa ajetaan asiakkaan testiympäristössä oman koneen sijaan. Opin tässä lisää hyvää rutiinia provisiontiin liittyen.

Viikkoraportti

Päällimmäisenä viikosta jäi mieleen työsopimuksen allekirjoittaminen. Jotain olen varmasti onnistunut tekemään oikein, kun yhtiössä päätettiin minun palkkaamisestani siitä huolimatta, että tämänhetkinen maailmantilanne on omiaan lisäämään epävarmuutta resurssien tarpeesta tulevaisuudessa. Vaikuttaisi kuitenkin siltä, että ainakin lyhyellä tähtämellä pandemian vaikutus Alfamen toimintaan on keskimääräistä pienempi. Varsinkin julkisen sektorin asiakkaat ovat suhteellisen vakaita ja niiden integraatiotarpeet eivät tästä olennaisesti vähenny. Oman oppimiseni kannalta olen ottanut ensimmäisiä askeleita sitä kohti, että voisin tulevissa projekteissa tarvittaessa osata toteuttaa myös projektin provisiontiin liittyvät tehtävät. Projektissa käytetään Ansiblea Docker-konttien luontiin ja käynnistämiseen. Tämän ymmärtämiseen on ollut apua käymästäni Ansible-verkkokurssista (Lunsford, C. 2015. Mastering Ansible) ja Docker-verkkokurssista (John Thompson. 2020. Docker for Java Developers). Dockerin ja Ansiblen yhdistämiseen liittyen katson mallia aiemmista projekteista ja niiden Ansible-playbook:sta. Epäselvissä tilanteissa voin kysyä neuvoa projektin arkkitehdiltä.

Tässä vaiheessa epäselvyyksiksi voidaan lukea salaus ja tietoturvakäytännöt. Näihin on tarkoitus palata ensi viikolla.

Seurantaviikko 9

Maanantai

Provisionti-inventaarioiden luonti jatkuu tänään prod- eli tuotantopuolen muuttujilla. Lisäksi salataan salattavaksi tarkoitetut salasanat. Salaukseen käytetään Ansiblen vault-toimintoa. Tähän löysin dokumentaatiota Ansiblen kotisivulta (Ansible. 2020. Kotisivu). Salaus muuttaa salasanan inventaariossa selkokieelisestä kryptatuksi ja se suojataan vaultin salasanalla, jota ei tallenneta mihinkään tiedostoon. Tällä tavalla salataan erinäisten palvelinten salasanoja. Komennot opittuani salasanojen salaaminen oli sangen suoraviivainen prosessi. Komentorivillä annetaan ansible-vault-komento käyttäen kunkin palvelun nimeä ja salasanaa. Tuloksena saadaan kryptattu salasana, joka asetetaan inventaarioon. Kun kryptaus tehdään oikealla salasanalla, osaa Ansible purkaa salauksen automaattisesti, mutta ihminen ei sitä pääse lukemaan. Tätä aihetta oli sivuttu tekemässäni Ansible-verkkokurssissa, mutta nyt pääsin kertaamaan tätä oikean projektin yhteydessä.

Tänään sain vaihteeksi myös huonoja uutisia. Projektin arkkitehti vaihtaa noin kolmen viikon päästä työpaikkaa. Toki mukavaa hänelle, mutta itse pohdin huolestuneena, kuka minua sitten auttaa ja perehdyttää jatkossa harjoittelujaksoni loppupuolella.

Tiistai

Tämä päivä alkaa edellisen projektin muutoksiin liittyvällä ongelmanselvityksellä. Projektissa piti muuttaa pari kohtaa sen suhteen, mitä tietoa tallennetaan mihinkin kenttiin. Tämän jälkeen havaitsin, että sähköpostinlähetykseen liittyvä testi projektissa ei mene läpi, vaikka tämän ei olisi pitänyt mitenkään muuttua tehtyjen muutosten yhteydessä. Kollegalla sama testi kuitenkin meni läpi. Palasin ajatuksissani seurantaviikolla kolme tutkimiini asioihin ja päädyin lopulta kasvattamaan prosessin odotusaikaa sähköpostin lähetyksen yhteydessä. Tämä ratkaisi ongelman. Muutosten ja korjausten jälkeen sovellus piti vielä asentaa uudestaan asiakkaan testiympäristöön. Tämä tehtiin tällä kertaa siten, että projektin arkkitehti neuvoi minua samalla kun minä suoritin asentamiseen liittyvät komennot ja kirjoitin niitä ylös Alfamen sisäiseen ohjesivustoon, jotta tähän olisi jatkossa ohjeet löydettävissä minulle ja muille uusille tekijöille.

Asennuksen jälkeen prosessi käynnistettiin prosessimanagerilla. Opin tässä suorittamaan asennuksen testiympäristöön ja testaamaan toimivuutta seuraamalla kontin sisäistä log-tietoa. Lisäksi opin käyttämään sisäistä ohjesivustoa ja päivittämään sinne ohjeita, jotta kerran opittu asia olisi muidenkin uusien tekijöiden löydettävissä.

Keskiviikko

Tänään tehdään uuteen projektiin viimeistelyä. XML:n keräyksestä puuttui edelleen pari tietokenttää, joita ei ollut sinne lisätty. Kentät puuttuivat, koska niitä ei ollut myöskään lähdetietona käytetyssä XML-tiedostossa. Nyt kentät lisättiin. Sain arkkitehdiltä neuvoja muun muassa siitä, kuinka päivämäärämuotoista tietoa käsitellään Pythonissa. Tähän asti kaikki projektissa tuotettu Python-skripti on toiminut aivan samalla tavalla kuin vertailun vuoksi vaikkapa PyCharmilla käännetty tavallinen Python-koodi toimisi. Normaali Pythonin tapa käsitellä päivämääriä ei kuitenkaan vaikuttanut nyt toimivan odotetusti.

Olen käyttänyt tähän mennessä termiä Python, koska sekä Python että Jython ovat minulle olleet yhtä outoja. Tässä vaiheessa aloin kuitenkin oppia joitakin eroja tavallisen Pythonin ja tässä projektissa käytetyn Pythonia paljon muistuttavan erikoistapauksen - Jythonin - välillä. Esimerkiksi sijoitettuani ensin datetime-tyyppisen tiedon muuttujaan en voinut enää käyttää siihen strftime-metodia, koska datetime-tyyppinen tieto muuttuikin yllättäen java.sql.Timestamp -muotoon muuttujaan sijoitettaessa. Etsin Googlen kautta tietoa tästä ja löysin hyvän tavan ongelman kiertämiseen Performance Engineering Solutions -sivustolta (Performance Engineering Solutions. 2019. Jmeter_python_gotchas: Datetime objects). Osoittautuu, että objekti pysyy Python datetime -muotoisena niin kauan, kun tietoa käsitellään yhdellä ja samalla rivillä ennen sen sijoittamista muuttujaan. (Liite 13)

Torstai

Tänään on vaihteeksi luvassa ainakin kaksi mielenkiintoista palaveria. Ensimmäisessä eräs kollegoistani esittelee oman lopputyönsä ja toisessa minulle perehdytetään minulle ennestään vieraan toisen asiakkuuden asioita. Tarkoitus on, että jatkossa työskentelen myös tämän toisen asiakkuuden projektien parissa. Kyseessä on kohtalaisen monimutkainen kokonaisuus ja suurehko julkisen sektorin asiakas, jolla on paljon kriittisiä toimintoja, joiden suhteen täytyy olla erityisen huolellinen.

Ensimmäisessä palaverissa kollega esitteli omaa lopputyötään ja kyseisen tuotteen käyttöliittymää päästiin myös hieman kokeilemaan palaverin aikana. Vaikutti hyvältä. Uuden asiakkuuden suhteen perehtyminen aloitettiin pienellä yleiskatsauksella ja kehitysympäristön pystyttämällä. Aiemmasta asiakkuudesta poiketen tässä ei käytetä Dockeria. Sen sijaan asensin Virtualbox-koneita käyttäen

Vagrantia, jolla virtuaalikoneita saa päälle ja pois päältä aina samalla itse määritellyllä tavalla käyttäen erityistä Vagrantfile-tiedostoa. Opin tässä hieman vaihtoehtoista tapaa kehitysympäristön pystyttämiseen ja sain uusiin asiakasprojekteihin valmistautumisen hyvälle alulle kyseisen asiakkuuden parissa aiemmin työskennelleen kollegan avustuksella. Kehitysympäristön pystytys tehtiin siten, että minä suoritin komentoja kollegan neuvojen perusteella samalla kun jaoin näyttöä hänelle ja toiselle kollegalle, joka työskentelee samassa projektissa kanssani. Tämä oli hyvä tapa päästä tässä asiassa alkuun.

Tämänhetkisen projektin puolella tehtävänäni oli selvittää, miksi viimeisimmän projektin ympäristömuuttajat eivät luotettavasti tallennu Docker-kontissa application.properties-tiedostoon. Tutkin tätä vertaamalla nykyisen ja aiempien projektien Ansible-playbook-tiedostoja. Havaitsin, että inventaarioiden prioriteetti oli tässä projektissa erilainen ja tietyssä tilanteessa yhden inventaariokansion sisältö ylikirjoitti virheellisesti korkeimmalle prioriteetille tarkoitetun inventaariokansion tiedostossa olevia arvoja. Muutin prioriteetin tarkoituksenmukaisemmaksi, jonka jälkeen projektin arkkitehti raportoi ongelman poistuneen. Tässä minulla oli kyllä ehkä lähinnä onnekas arvaus siitä, mistä ongelma voisi johtua ja se osui oikeaan. Voi tätä kuitenkin pitää onnistumisena, kun keksin nopeasti ratkaisun, jota projektin arkkitehti ei ollut heti muilta kiireiltään keksinyt.

Viikkoraportti

Tälle viikolle kertyi poikkeuksellisesti vain neljä työpäivää. Kaikista onnistumisista huolimatta myönnettäköön, että viikkoa leimasi minulla pienimuotoinen muutoksesta johtuva hämmennys. Olin juuri ehtinyt päästä hyvään yhteistyöhön projektissa työskennelleen kollegan ja projektin arkkitehdin kanssa, mutta nyt lyhyen ajan sisällä molemmat vaihtavatkin työpaikkaa. Tämä luo toisaalta minulle mielenkiintoisen sopeutumishaasteen. Minun tulee nyt näyttää, että pystyn työskentelemään yhtä hyvin muidenkin henkilöiden kanssa ja että tähänastinen oppimiseni ei ole ollut täysin näiden kahden henkilön avuliaisuuden varassa. Toisaalta voin pitää positiivisena asiana sitä, että olen pääsemässä laajentamaan osaamistani muiden asiakkuuksien ja projektien puolelle. Voidaan myös ajatella, että olen saamassa lisää vastuuta, mitä taas voidaan pitää luottamuksenosoituksena työnantajan puolelta. Nyt saan osoittaa olevani tämän luottamuksen arvoinen. Epäilemättä perehtyminen toiseen asiakkuuteen kestää joitakin viikkoja, koska joudun totuttelemaan uudenlaiseen tiimiin ja eri tavalla suunniteltuun projektiin. Aion tässä toteuttaa avoimuuden periaatetta. Kysyn, jos jokin asia on epäselvä ja myönnän avoimesti, jos jokin

asia jäi ymmärtämättä. Olen edelleen harjoittelija, joten koen, että minulla on yhä myös oikeus olla osaamatta heti kaikkea. Toisaalta nyt on tärkeää edelleen pyrkiä oppimaan ja sisäistämään asioita nopeasti, jotta olen mahdollisimman valmis tehokkaaseen työskentelyyn mieluiten jo ennen varsinaisen työsopimuksen alkua.

Yhtenä käytännön asiana, johon aion kiinnittää huomiota uuteen projektiin ja uudenlaiseen tiimiin siirryttäessä, on oman koodini luettavuus. Tässä aion edelleen käyttää hyödyksi Clean Code-kirjan oppeja liittyen koodin asetteluun, jäsentelyyn ja muuttujien ja metodien nimeämiseen, unohtamatta kysyä asiaan liittyen kommentteja myös kollegoilta, jotta käytäntömme on varmasti yhtenäinen.

Seurantaviikko 10

Tiistai

Viikko alkaa tällä kertaa tiistaista, koska maanantai oli vapaapäivä. Tänään on tarkoitus jatkaa torstaina alkanutta perehtymistä minulle uuteen asiakkuuteen ja jatkamme siihen liittyvän kehitysympäristön pystyttämistä. Todennäköisesti viimeksi käyttöönotettuun virtuaaliympäristöön asennetaan seuraavaksi sovelluksia.

Tässä toisessa asiakkuudessa eri integraatioprojektit nivoutuvat toisiinsa hieman eri tavalla kuin minulle aiemmin tutuksi tulleen asiakkaan projekteissa on ollut tapana. Tätä varten ennen kehitystyön aloittamista, asennetaan virtuaalikoneille sovelluksia, jotka liittyvät kyseisen asiakkuuden aiempiin projekteihin, joissa muun muassa tietyt tietokannat vaikuttaisivat olevan käytössä useammassa projektissa asiakkuudella. Tämän jälkeen uutta projektia voidaan aloittaa rakentamalla uutta koodia näiden aiempien projektien rinnalle. Sovellukset asennetaan käyttäen ansible-playbook-tiedostoja.

Suurin osa ansible-playbook-suorituksista onnistui, mutta muutamasta tuli erilaisia virheitä. Playbookien suorittaminen myös kesti yllättävän kauan. Tätä jatketaan huomenna. Oman oppimisen kannalta tässä näin hyötynä sen, että eri ansible-playbook-tiedostoja suorittamalla ja tutkimalla pääsen

pikkuhiljaa tutustumaan minulle aiemmin tuntemattoman asiakkuuden projekteihin ja niiden osasiin. Haasteellista tässä on kokonaisuuden massiivisuus, joten hahmottaminen on aloitettava pieninä palasina.

Keskiviikko

Tänään selvitän vielä ongelmaa edellisen projektin yhteyshenkilön haun kanssa. Havaittiin, että yhteyshenkilön haku ei tietyssä tilanteessa onnistunut, vaikka haussa käytetyn lähdemateriaalin perusteella yhteyshenkilön tulisi kyllä löytyä.

Käytin selvityksessä apuna Insomnia-sovellusta ja muutin hakuehtoja hieman. Syy löytyi siitä, että haku toimii oikein, mutta yhteyshenkilön sähköpostiosoite oli asetettu päätteeltään määritysten vastaisesti. Päädyttiin siihen, että asiasta informoidaan asiakasta, joka korjaa tiedot tietokannassaan tarvittaessa määritysten mukaisesti tai vaihtoehtoisesti voidaan sopia määritysten muuttamisesta.

Eilinen ansible-playbookien ongelma ratkesi osittain sillä tavalla, että tutkin virtuaalikoneelle tallentunutta sovelluksen log-tiedostoa. Tästä ilmeni, että ainakin kahdessa tapauksessa virtuaalikoneelle määritetty muistin määrä oli loppunut kesken. Korjasin tämän kasvattamalla virtuaalikoneelle määritetyn muistin määrää Vagrantfile-tiedostossa kolmesta kahdeksaan gigatavuun. Toinen virhe taas aiheutui siitä, että sovellus virtuaalikoneella ei saanut yhteyttä toisella virtuaalikoneella sijaitsevaan palvelimeen. Tämä yhteysongelma jää selvitettäväksi huomenna.

Pääsin myös auttamaan toista kollegaa kehitysympäristön pystyttämisessä myös hänelle. Kävimme läpi samoja virtuaalikoneiden käyttöönottoon liittyviä asioita, joita itse opettelin eilen. Haasteena tässä oli muun muassa Ansiblen ja Pythonin oikeiden versioiden löytäminen ja käyttöönotto. Tähän liittyen saimme vielä apua kolmannelta kollegalta videoneuvottelun kautta.

Opin tässä kuitenkin jälleen ongelmanratkaisua ja sain selville muun muassa ansible-playbookin virheilmoitusten ja log-tiedostojen tutkimiseen liittyvää logiikkaa. Tässä minua auttoi videoneuvottelun kautta kyseisen asiakkuuden parissa aiemmin työskennellyt kokenut kollega.

Torstai

Uuden asiakkuuden ansible-playbook-tiedostoihin liittyvien ongelmien ratkaisua jatketaan vielä tänään. Toivottavasti tänään päästään myös tekemään koodausta tähän liittyen, mahdollisesti aloitetaan jokin pieni uusi integraatio tai tehdään jokin pieni muutos aiempaan koodiin.

Työnantajaa vaihtavan arkkitehtimme tilalle kaavailtu uusi arkkitehti onnistuikin tänään auttamaan meitä sen verran, että saimme kehitysympäristön pystytyksessä kohdatut ongelmat ratkaistua. Ratkaisun tärkeimpänä kohtana löysimme projektin kuormanjakajan testauksessa ongelman, joka oli saanut alkunsa hiljattain tehdystä muutoksesta. Muutimme Nginx:n konfiguraatiota siten, että testi saatiin menemään läpi. Tämän jälkeen suoritimme ansible-playbookit läpi uudestaan kohtaamatta enää uusia virheitä.

Päivän loppupuolella pääsimmekin jo koodaamaan pienen muutoksen järjestelmään, jossa viestejä käsitellään useissa eri jonoissa. Lisäsimme yhden uuden jonon ja määrittelimme ehdot, joilla viestejä siirretään kyseiseen jonoon viestin tyyppin perusteella. Testasimme myös jonon purkamista luomalla testiviestejä ja luomalla tilapäisen palvelimen, joka sitten haki näitä viestejä jonosta. Kyseisen asiakkuuden parissa aiemmin työskennellyt kollega kertoi videoneuvottelussa juuri tarvittavan määrän tietoa tämän muutoksen toteuttamiseksi haastaen ja kysyen samalla pieniä päättelykykyä vaativia asioita siten, että oli mahdollisuus myös asioiden oivaltamiseen. Tällä tavoin asiakkuus, siihen liittyvät sovellukset ja järjestelmät, ja niitä käyttävät projektit alkavatkin hahmottua pikkuhiljaa asia kerrallaan. Hyvänä virstanpylväänä voikin pitää sitä, että sain työpäivän lopuksi tehtyä ensimmäisen commitin tämän asiakkuuden git-repositoryyn.

Perjantai

Tämän päiväkirjan viimeinen seuranta päivä alkaa hyvistä tunnelmista. Nyt on vihdoinkin alkanut vahvistua käsitys siitä, että myös tämän uuden asiakkuuden tekeminen on mahdollista ottaa haltuun kohtalaisen ajan kuluessa. Tänään kävin läpi listausta tehtävistä, jotka meidän on tarkoitus seuraavaksi suorittaa tähän asiakkuuteen liittyen. Uusia integraatioita ei nyt kuitenkaan tänään lähdetä enää toteuttamaan sen vuoksi, että tälle viikonlopulle on aikataulutettu tuotantoon vienti ja se halutaan suorittaa ensin, ennen kuin tehdään lisää muutoksia. Iltapäivän kahvitauolla sain esittää kiitoksia kollegoille hyvin alkaneesta perehdytyksestä ja avusta kehitysympäristön pystyttämisessä ilmenneisiin ongelmiin. Tehtävälistaa läpikäydessäni sain taas tietää lisää asiakkuuteen liittyvien integraatioiden paljoudesta ja näihin liittyvien haasteiden laadusta. Enimmäkseen kyseessä on tarve muutamalle melko suoraviivaiselle ja pienelle muutokselle, mikä on ilmennyt asiakkaalle palveluita käytettäessä. Esimerkiksi yksinkertaisesta

muutoksesta tuotannossa olevaan integraatioon käyköön tiedonkäsittelyn yhteydessä lähetettävän viestin otsikon muuttaminen. Ensi viikolla päästään varmasti toteuttamaan useampikin tällainen pieni muutos.

Viikkoraportti

Edellisen viikon hämmennys ja huoli on tällä viikolla vaihtumassa vahvistuneeseen tunteeseen siitä, että kyllä täällä vielä riittää erinomaisia osaajia ja hyviä perehdyttäjiä siitäkin huolimatta, että aluksi muun muassa tutuksi tullut arkkitehti talosta poistuu. Lisäksi tekeminen on monipuolistumassa ja vaikuttaa siltä, että minulla on mahdollisuuksia päästä vielä aiempaa paremmin tilanteeseen, jossa vastuuta riittää ja tekeminen ei pääse loppumaan yllättäen kesken. Kehitysympäristön käyttöönotossa ilmenneet ongelmat saatiin viikon aikana ratkaistua. Haasteellisinta tämän ratkaisussa oli tällä kertaa kohdattujen virheilmoitusten ainutlaatuisuus ja heikohko kuvaavuus verrattuna siihen, mitä toimenpiteitä virheen korjaaminen vaati. Tällä kertaa ongelman ratkaiseminen esimerkiksi Google-hauilla ei ollut mahdollista, koska käytetty järjestelmä ei ole tarpeeksi yleinen, jotta monikaan satunnainen henkilö maailmalla olisi kohdannut vastaavia virheilmoituksia. Tällöin paras tapa ongelman ratkaisemiseen olikin jälleen suora apu kokeneemmilta kollegoilta videoneuvottelun kautta.

Yksi hyväksi havaittu tapa kehittää omaa osaamista on samalla kerätä suoritettujen komentojen talteen itselle tai sisäiseen ohjesivustoon, jotta seuraavalla kerralla muistetaan, mitä on tehty.

4 POHDINTA

Tätä päiväkirjaa kirjoittaessani olen kehittynyt nykytilanteen kuvaus -kohdassa mainitun aloittelevan toimijan tasolta kohti syvällisempää ymmärrystä työtehtävien suorittamisesta. Kykenen aiempaa useammin ratkaisemaan minulle annettuja tehtäviä itsenäisesti, ja vaikka kysynkin silloin tällöin apua kollegoilta, voin myös sanoa yhtä usein vastaavasti auttavani muita heitä askarruttavissa asioissa. Käymäni verkkokurssit ovat kasvattaneet osaamistani, mutta ennen kaikkea käytännön koodauksen harjoittelu ja projekteissa mukanaolo ovat tuoneet näkemystä teoriassa opittujen asioiden käyttöönnottoon.

Asiakasprojekteissa olen oppinut olemaan mukana Scrum-tiimissä. Päivittäisissä Scrum-palavereissa on käyty läpi edelliset ja seuraavat tehtävät ja pureuduttu mahdollisiin toiminnan esteisiin ja niiden poistamiseen. Planning-palavereissa olen ollut mukana joka toinen viikko suunnittelemassa seuraavan Sprintin tekemisiä. Mielestäni tämä toimintamalli on ollut parhaimmillaan oikeinkin tehokasta ja mielekästä, varsinkin kun yhteistyö kollegoiden kanssa on sujunut mainiosti.

Ohjelmistokehitys Mac-koneella on ollut minulle aiemmin täysin tuntematon käsite, mutta nykyään tämä aiemmin hyvin vähän käyttämäni ympäristö on muodostunut tutuksi. Olen oppinut paljon keskeisiä asioita rajapinnoista ja palvelinten välisestä kommunikaatiosta. Tiedon hakeminen palvelimilta REST-asiakasohjelmistojen avulla on tullut tutuksi ja tällä tavalla varmistetusti toimivien tiedonhakujen

löytäminen on auttanut vastaavan tiedon hakemista Mule-flown kautta. Olen myös kasvattanut osaamistani Unix-komentojen ja komentorivi-terminaalin käytössä. Muun muassa Vagrant-, Ansible-, Docker- ja Maven-komentojen käyttö on tullut hyvin tutuksi päivittäin. Tässäkin mielessä MacOS:n käyttö ohjelmistokehityksessä muistuttaa paljolti Linux-ympäristöä.

Yksi hyvä tapa, jonka olen löytänyt uusien asioiden hahmottamiseen, on käsittelyssä olevan tiedon ja erikseen sen tyyppin lukeminen log-ruudulta tai debuggerista ennen, jälkeen ja kesken sillä hetkellä tutkimaani koodin osaa tai metodia. Esimerkiksi tutkinnassa olevan muuttujan tietotyypin tarkistaminen login avulla on osoittautunut tärkeäksi keinoksi ongelman selvittämisessä ja tietorakenteen tutkimisessa askel kerrallaan.

Yksi mielenkiintoinen toimintatapa, joka on omassa tiimissäni otettu käyttöön näiden projektien aikana, on Jython-skriptikomponenttien käyttö osana Mule-floweja. Ymmärtääkseni näitä ei ole juuri muissa tiimeissä käytetty, ja tässä minulla onkin tilaisuus tuoda talossa laajemmin tietoon tällainen hyvä työkalu. Tällä tavoin voi projektista riippuen olla helpompaa tehdä tiettyjä muutoksia ja jäsentelyjä käsiteltävään tietoon. Mahdollisesti voin tästä saada lähitulevaisuudessa aiheen ensimmäiselle itse järjestämälleni lyhyelle lounasvalmennukselle, mikäli tällainen koetaan tarpeelliseksi.

Tämän päiväkirjan kirjoittamisen aikana olen ennen kaikkea oppinut tekstimuotoisen Mule-flown käsittelyä ja tiedonkäsittelyä Mulen ja Activitin välillä. Olennaisinta oli oppia ymmärtämään ne erilaiset tavat, joilla Mule flowssa lähetetään ja vastaanotetaan tietoa. Kun tiedon voi välittää Mule-flowsta Java- tai Python-komponentille ja palauttaa sieltä takaisin halutussa muodossa, voi tiedolle käytännössä aina tehdä kaikki tarvittavat muutokset.

Päiväkirjan kirjoittaminen on auttanut selkeyttämään omaa ajatusmaailmaa ja kokonaisuuden hahmottamista tämän työn aloittamiseen liittyen. Olen myös kahteen otteeseen etsinyt ja löytänyt päiväkirjamerkintöjen avulla ratkaisuja ongelmiin ja haasteisiin, jotka ovat toistuneet toisen projektin parissa, mikä on nopeuttanut ongelmanratkaisua toisella kerralla.

Ammatillisen kehittymiseni kannalta seuraavat askeleet minulla ovat varsinaisen työsuhteen aloittaminen, virallisen Mule-sertifikaatin suorittaminen ja verkkokurssit Springistä ja Boomista sekä mahdollisesti Reactista ja Angularista. Kirjallisuuden puolelta on tarkoitus perehtyä Dockerista kertoviin kirjoihin ja yleisemmin Scrum- ja DevOps-toimintamalleista kertoviin kirjoihin.

Tämän päiväkirjan kirjoittamisen aikana varmistui myös, että harjoittelusopimuksella aloittamani työnteko jatkuu kesällä vielä harjoittelun päättymisen jälkeen toistaiseksi voimassa olevan

työsopimuksen merkeissä. Näin ollen minulla on nyt hyvät näkymät siihen, että pääsen kehittämään omaa osaamistani pitkäjänteisesti ja hyvillä mielin, mielenkiintoisessa työpaikassa.

LÄHTEET

Alfame. 2020a. Integraatio-opas. Saatavissa:

https://www.alfame.com/hubfs/files/Integraatio-opas_sujuvaan_tietojarjestelmaintegraatioon.pdf .

Viitattu 5.2.2020

Alfame 2020b. Kotisivu. Saatavissa: <https://www.alfame.com/> . Viitattu 5.2.2020

Ansible 2020. Kotisivu. Saatavissa: https://docs.ansible.com/ansible/latest/user_guide/vault.html .

Viitattu 6.4.2020

Boomi. 2020. Kotisivu. Saatavissa: <https://boomi.com/services/training/>

Lunsford, C. 2015. Mastering Ansible. Saatavissa:

<https://www.udemy.com/course/mastering-ansible/> . Viitattu 6.4.2020

Codota. 2020. Best Javacode snippets using

org.mule.runtime.core.api.transformer.TransformerException. Saatavissa:

<https://www.codota.com/code/java/classes/org.mule.runtime.core.api.transformer.TransformerException> . Viitattu 23.3.2020

Karim, W. 2020. Converting JSONObject to HashMap. Saatavissa: osoitteessa

<https://wajahatkarim.com/2018/11/converting-jsonobject-to-hashmap/> . Viitattu 17.2.2020

Digitalocean. 2020. Kotisivu. Saatavissa: <https://www.digitalocean.com> . Viitattu 19.3.2020

Git-scm. 2020. Download objects and refs from another repository. Saatavissa:

<https://git-scm.com/docs/git-fetch> . Viitattu 12.3.2020

Moilanen, P. 2020 Git. Spring-boot-docker-1 git repository. Saatavissa:

<https://github.com/Perttu219/spring-boot-docker-1> . Viitattu 13.2.2020

Udemy. 2019. Python for Absolute Beginners. Saatavissa:

<https://www.udemy.com/course/python-for-absolute-beginners-u/> . Viitattu 2.3.2020

Crunchify. 2020. How to Read JSON Object From File in Java? Saatavissa:

<https://crunchify.com/how-to-read-json-object-from-file-in-java/> . Viitattu 17.2.2020

Insomnia. 2020. Insomnia REST client. Saatavissa: <https://insomnia.rest/> . Viitattu 25.3.2020

Thompson, J. 2020. Docker for Java Developers. Saatavissa:

<https://www.udemy.com/course/docker-for-java-developers/> . Viitattu 13.3.2020

Mulesoft LLC. 2020. HTTP Request Configuration Reference:

<https://docs.mulesoft.com/connectors/http/http-request-ref> . Viitattu 17.2.2020

Mulesoft LLC. 2019 Flows and Subflows. Saatavissa:

<https://docs.mulesoft.com/mule-runtime/4.2/about-flows> . Viitattu 10.2.2020

Campbell, B. 2019. OAuth 2.0 Mutual TLS Client Authentication and Certificate-Bound Access Tokens. Saatavissa: <https://tools.ietf.org/id/draft-ietf-oauth-mtls-13.html> . Viitattu 28.2.2020

- Wikipedia. 2020. Ohjelmointirajapinta.Saatavissa:
<https://fi.wikipedia.org/wiki/Ohjelmointirajapinta> . Viitattu 8.2.2020
- Performance Engineering Solutions. 2019. Jmeter_python_gotchas: Datetime objects. Saatavissa:
<https://performance-engineering-solutions.com/2019/03/06/jmeter-python-gotchas/> . Viitattu 8.4.2020
- Martin, R. 2009. Clean Code: A handbook of agile software craftsmanship.
15. painos. 2009. Westford, Massachusetts. Pearson Education, Inc.
Viitattu 6.3.2020 ja 10.4.2020
- Thompson, J. 2015. Springframeworkguru git repository. Saatavissa:
<https://github.com/springframeworkguru/spring-core-spring-mvc.git>
Viitattu 13.3.2020
- Stack Overflow. 2020. Kotisivu. Saatavissa:
<https://stackoverflow.com/questions> . Viitattu 17.2.2020
- Stack Overflow. 2020. How to throw a custom exception from inside Java Transformer in Mule ESB 3.8?
Saatavissa: <https://stackoverflow.com/questions/54360843/how-to-throw-a-custom-exception-from-inside-java-transformer-in-mule-esb-3-8> . Viitattu 23.3.2020
- Buchalka, T. Lochert, G. 2019. Java Programming Masterclass for Software Developers.
<https://www.udemy.com/course/java-the-complete-java-developer-course/> . Viitattu 10.2.2020
- Stack Overflow. 2020. Viewing unpushed git commits. Saatavissa:
<https://stackoverflow.com/questions/2016901/viewing-unpushed-git-commits> . Viitattu 12.3.2020
- W3Schools. 2020. Python isinstance() Function. Saatavissa:
https://www.w3schools.com/python/ref_func_isinstance.asp . Viitattu 26.3.2020.

Liitteet

Esimerkki kuinka JSON-tiedosto luetaan ja palautetaan hashmappina:

```
public class ParseParametersFromFile {
    @SuppressWarnings("unchecked")

    public static Map parameters(String fileLoc)
    {
        JSONParser parser = new JSONParser();
        try {
            Object obj = parser.parse(new FileReader(fileLoc));

            JSONObject jsonObject = (JSONObject) obj;

            String jsonString=jsonObject.toJSONString();
            Map<String, String> mapObj = new Gson().fromJson(
                jsonString, new TypeToken<HashMap<String, Object>>() {}.getType());

            return mapObj;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

LIITE 2

Esimerkki kuinka dataa tuodaan Muleen requiredPropertiesilla ja viedään commitPropertiesilla:

```
<spring:property name="requiredProperties">
  <spring:map>
    <spring:entry key="fooBar">
      <spring:value>fooBar</spring:value>
    </spring:entry>
  </spring:map>
</spring:property>

<spring:property name="commitProperties">
  <spring:map>
    <spring:entry key="concreteLeadJson">
      <spring:value>concreteLeadJson</spring:value>
    </spring:entry>

  </spring:map>
</spring:property>
```

Esimerkissä haetaan Pythonilla Jsonista toteutuskoodi.

```
import json

try:
    concrete = json.loads(concreteJson)
    result = concrete['productInformation']['implementationCode']
except KeyError:
    result = dynamicsToteutusDefaultValue
```

Esimerkissä käynnistetään for-loopissa prosessi-instansseja, joihin on sisällytetty haluttua dataa variables HashMapin sisällä. Prosessi-instansseja käynnistetään yksi jokaiselle ilmoittautumis-ArrayLististä löytyvälle ilmoittautumiselle.

```
for (int i = 0; i < enrollments.size(); i++) {

    Map<String, Object> variables = new HashMap<>();
    variables.put("mule_invocation_concreteJson", gson.toJson(enrollments.get(i)));

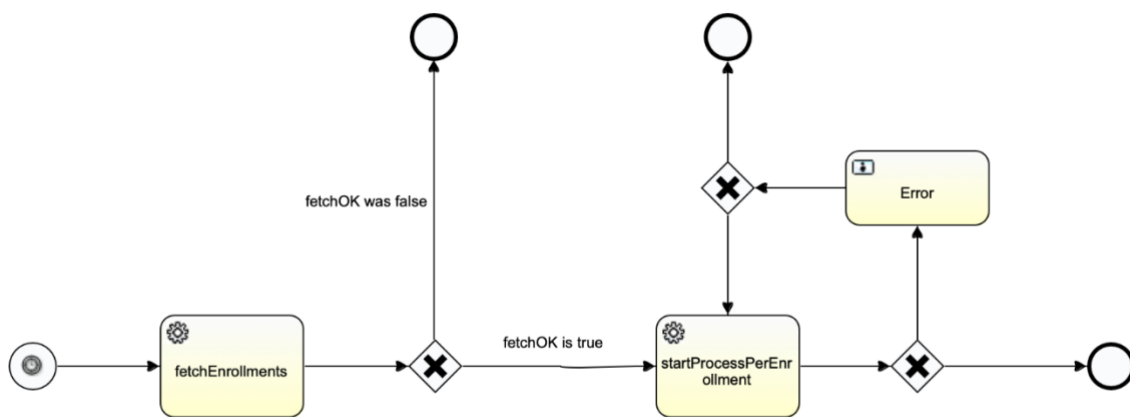
    LinkedHashMap temp = (LinkedHashMap) enrollments.get(i);
    String enrollmentGuid = temp.get("enrollmentGuid").toString();
    variables.put("mule_invocation_enrollmentGuid", enrollmentGuid);

    ProcessFactory.startProcessInstance(runtimeService, processDefinationKey, null, variables,
tenantId);

}
```

LIITE 5

Yksinkertainen Activiti-prosessikaavio. Lomakkeita haetaan 5 minuutin välein ja käynnistetään jokaisesta haetusta lomakkeesta seuraava Activiti-prosessi.



LIITE 6

Esimerkissä haetaan tuotekoodi Mule-flow:ssa python skriptin avulla:

```

<enricher target="#[flowVars.tuotekoodi]">
  <processor-chain>
    <set-payload value="#[flowVars.dynamicsToteutusJson]" />
    <scripting:component doc:name="Script">
      <scripting:script engine="jython" file="jython/parseTuoteKoodi.py" />
    </scripting:component>
  </processor-chain>
</enricher>
  
```

LIITE 7

Python parseri tuotekoodin hakemiseen

```
#parseTuoteKoodi.py
import json

try:
    dynamics = json.loads(payload)
    code = dynamics['_di_tuoteid_value']
    result = code.zfill(36)
except KeyError:
    result = dynamicsTuoteDefaultValue
```

LIITE 8

Esimerkissä tuoteId vietään payloadiin suoraan Mule-flow:ssa ilman python skriptiä:

```
<enricher target="#[flowVars.tuoteId]">
  <processor-chain>
    <set-payload value="#[flowVars.dynamicsToteutusJson]" />
    <json:json-to-object-transformer returnClass="java.util.Map" />
    <set-payload value="#[payload['_di_tuoteid_value']]" />
  </processor-chain>
</enricher>
```

LIITE 9

```
campus['Koulutus_Verkko-opiskelu']=False
```

```
vars= {}
vars = edu.get('Opiskelumuodot', {}).get('Opiskelumuoto', [])
for method in vars:
    if (isinstance(method, dict) and method.get('Nimi')== 'Verkko-opiskelu') or \
        (isinstance(method, unicode) and vars.get('Nimi')== 'Verkko-opiskelu'):
        campus['Koulutus_Verkko-opiskelu']=True
```

LIITE 10

Tämä esimerkkiohjelma tulostaa annetun päivämäärän 'Jun 09,2019' muodossa: 2019-06-09:

```
import datetime
d = datetime.datetime.strptime('Jun 09,2019', '%b %d,% Y').strftime('%Y-%m-%d')
print(d)
```

LIITE 11

Esimerkissä SNMP/SMTP stub

```
<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:smtp="http://www.mulesoft.org/schema/mule/smtp"
  xmlns:spring="http://www.springframework.org/schema/beans"
    http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-current.xsd
  http://www.mulesoft.org/schema/mule/smtp
http://www.mulesoft.org/schema/mule/smtp/current/mule-smtp.xsd

  <spring:beans>
    <spring:bean id="smtpStub" class="com.alfame.esb.utils.testing.SmtpServer"
      init-method="initialize" destroy-method="close">
      <spring:property name="port" value="{com.alfame.esb.smtp.port}" />
      <spring:property name="username" value="{com.alfame.esb.smtp.user}" />
      <spring:property name="password" value="{com.alfame.esb.smtp.password}" />
    </spring:bean>
  </spring:beans>
</mule>
```

Käyttöesimerkki:

```
@Autowired
```

```
@Qualifier("smtpStub")
```

```
SmtpServer smtpStub;
```

```
@ClassRule
```

```
public static DynamicPortProperties smtpPort = new DynamicPortProperties(
"com.alfame.esb.smtp.port" );
```

```
@Override
```

```
protected String getConfigResources() {
```

```
    return String.join(", ",
        "stubs/snmp-stubs.xml",
        "sendErrorEmail.xml",
```

```
    );
```

```
}
```

```
@Test
```

```
public void testSendErrorEmail() throws Exception {
```

```
    smtpStub.waitForMessages(30000,1);
```



```

MimeMessage[] messages = smtpStub.getReceivedMessages();
for( MimeMessage message : messages ) {
String msg = IOUtils.toString( message.getInputStream() );
logger.debug( message.getSubject() );
logger.debug( msg);
}

```

LIITE 12

Esimerkissä korjattu HTTP-request konfiguraatio

```

<http:request-config name="concreteLeadsRequestConfig"
    host="{ com.alfame.esb.projektin.host}"
    port="{ com.alfame.esb.projektin.port}"
    protocol="{ com.alfame.esb.projektin.protocol}"
    basePath="/"
    doc:name="concreteLeadsRequestConfig">
<http:basic-authentication username="{ com.alfame.esb.projektin.username}"
    password="{ com.alfame.esb.projektin.password}"
    preemptive="true"/>
<tls:context enabledProtocols="TLSv1.2"
enabledCipherSuites="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256">
    <tls:trust-store path="{ com.alfame.esb.tls.truststore.path}"
        password="{ com.alfame.esb.tls.password}"
        insecure="{ com.alfame.esb.tls.allowInsecure:false}"
        type="jks" />
</tls:context>
</http:request-config>

```

LIITE 13

Esimerkissä Python-skripti, jossa käytetty dict-tietorakennetta.

```

import json

vihannes_json= json.loads(vihannesLuetteloJson)

MAPPING = {
    'Peruna': '123',
    'Porkkana': '234',
    'Kaali': '345',
    'Kurkku': '456',
    'Lanttu': '567',
    'Parsa': '678',
    'Sipuli': '789'
}
esimerkki= {}
esimerkki['koodi'] = MAPPING.get(vihannes_json.get('vihannes'))

```

```
esimerkki['hinta'] = vihannes_json.get('eur_hinta')
#jne

result = json.dumps(esimerkki)
```

LIITE 14

Esimerkissä testiluokan addBuilders-metodi:

```
@Override
protected void addBuilders(List<ConfigurationBuilder> builders) {
    super.addBuilders(builders);
    builders.add(0, new ExtensionsManagerConfigurationBuilder());
}
```