



Sovelluskehitys asiakasprojekteissa low-code-teknologioilla

Sami Niskanen

Haaga-Helia ammattikorkeakoulu

Tradenomin tutkinto

Amk-opinnäytetyö

2022

Tekijä Sami Niskanen
Tutkinto Tradenomi
Opinnäytetyön nimi Sovelluskehitys asiakasprojekteissa low-code-teknologioilla
Sivumäärä 80
<p>Päiväkirjamuotoisessa opinnäytetyössä seurattiin tekijän ammatillisen osaamisen kehittämistä kahden low-code-teknologioita hyödyntävän asiakasprojektin kautta. Kahdeksan viikkoa kestänyt seurantajakso ajoittui aikavälille 29.8.2022–21.10.2022. Seuranta toteutettiin päivittäisillä päiväkirjamerkinnoilla. Viikon päätteeksi pääteema, tavoitteet ja toteuma yhdistettiin lähteitä hyödyntäen seurantaviikon päättäväksi viikkoanalyysiksi.</p> <p>Tekijä työskentelee suomalaisessa vuonna 2022 perustetussa ICT-konsultointia harjoittavassa yrityksessä. Tekijän työtehtävät koostuvat asiantuntijakonsultoinnista suomalaisille asiakkaille. Selvyyden vuoksi opinnäytetyöstä rajattiin kaikki muut työtehtävät pois ja raportissa keskityttiin ainoastaan kahden asiakasprojektin tehtäviin liittyvään seurantaan. Osaamisen kehittäminen tähtäsi low-code-teknologioilla tehtävän ketterän ohjelmistokehityksen teorian ja käytännön ymmärtämiseen, toteutukseen tarvittavien työkalujen teknisen ymmärryksen kasvattamiseen sekä asiakkaan kannalta oleellisten kehitystarpeiden tunnistamiseen ja priorisoimiseen.</p> <p>Pääosassa on POC-hanke, jonka tavoitteena on korvata asiakkaan tuotantokäytössä olevan ERP-järjestelmän oma mobiilisovellus. Mobiilisovellusta käytetään työ- ja päivystysajan leimaamiseen, työtilausten käsittelyyn, ostojen ja kulujen kirjaamiseen sekä kaluston raportointiin. Toinen opinnäytetyössä seurattava projekti on HR Häirintäilmoitus. Siinä tavoitteena on luoda HR-osastolle järjestelmä, jonka avulla HR voi seurata henkilöstön tekemiä häirintäilmoituksia, kirjata ilmoitusten selvitykseen liittyviä lisätietoja sekä analysoida datan perusteella mahdollisia ajankohtia, jolloin ilmoitusten määrä on kasvussa.</p> <p>Opinnäytteen tekijä on kehittynyt hyvin asetetuissa osaamisen kehittämisen tavoitteissa ja suurinta kehittyminen on ollut teknisen osaamisen osalta. Seurantajakson aikana tekijä on pystynyt kehittämään osaamistaan laajasti eri low-code-teknologioiden osalta. Opinnäytetyöprosessin alussa tekijän osaaminen rajoitti jonkin verran tapoja, joilla sovelluksia pystyttiin kehittämään mutta prosessin lopussa tekijä on pystynyt soveltamaan ja hyödyntämään seurantajakson aikana oppimia asioita kehityksessä. Ketterän kehityksen osalta kehittyminen oli vähäisintä, sillä seurantajakson aikana kummassakaan projektissa ei tullut sellaisia piirteitä, joiden kautta tekijä olisi päässyt syvemmin täydentämään aiempaa ketterän kehityksen ymmärrystään.</p> <p>Työtehtävien painopisteen keskittyessä asiakasprojekteihin, opinnäytteen seurantajakson aikana molempia projekteja on pystytty edistämään hyvin. Tämän myötä opinnäytetyöprosessista on ollut hyötyä myös asiakkaalle, sillä se on mahdollistanut jatkuvan edistymisen sovellusten kehityksessä. POC-hanke on testausvaiheessa, jonka jälkeen tavoitteena on pilotoida sovellusta asiakkaan loppukäyttäjien kanssa. HR Häirintäilmoitus on viimeistelyjä vaille valmis ja sen tavoiteltu käyttöönottopäivämäärä on 1.1.2023.</p>
Asiasanat sovelluskehitys, low-code, Power Apps, Power Platform, asiakasprojekti, konsultti

Sisällys

1	Johdanto	1
1.1	Työtehtävissä tarvittava osaaminen	2
1.2	Ammatillisen kehittymisen tavoitteet ja rajaukset	2
1.3	Tietoperusta	3
1.4	Keskeiset käsitteet	3
1.4.1	ICT-alan ammattikäsitteet ja asiakkaan käyttämä terminologia	4
1.4.2	Microsoftin tuotteisiin ja palveluihin liittyvät käsitteet	6
2	Lähtötilanteen kuvaus	8
2.1	Oman nykyisen työn analysointi	8
2.2	Sidosryhmien esittely	8
2.3	Työpaikan vuorovaikutustilanteet	10
3	Seurantajakson raportointi viikkoanalyysineen	11
3.1	Seurantaviikko 1 (vko 35)	11
3.2	Seurantaviikko 2 (vko 36)	17
3.3	Seurantaviikko 3 (vko 37)	23
3.4	Seurantaviikko 4 (vko 38)	29
3.5	Seurantaviikko 5 (vko 39)	38
3.6	Seurantaviikko 6 (vko 40)	45
3.7	Seurantaviikko 7 (vko 41)	51
3.8	Seurantaviikko 8 (vko 42)	60
4	Pohdinta	70
	Lähteet	75

1 Johdanto

Tämän päiväkirjamuotoisen opinnäytetyön tavoitteena on seurata osaamiseni kehittämistä asiakkaani kahden Power Apps projektin avulla. Osaamisen kehittäminen tähtää low-code-teknologioilla tehtävän ketterän ohjelmistokehityksen teorian ja käytännön ymmärtämiseen, toteutukseen tarvittavien työkalujen teknisen ymmärryksen kasvattamiseen sekä asiakkaan kannalta oleellisten kehitystarpeiden tunnistamiseen ja priorisoimiseen. Seuranta tehdään 29.8.2022 – 21.10.2022 välisenä aikana. Päiväkirjamerkinnot tehdään päivittäin ja niistä koostetaan jokaisen viikon kohdalla viikkoanalyysi, jonka tavoitteena on yhdistää viikon aikana opitut asiat eheäksi kokonaisuudeksi ja linkittää ne edellä mainittuihin osaamisen kehittämisen teemoihin ajankohtaisia lähteitä hyödyntäen.

Työskentelen tänä kesänä perustetussa ICT-alan konsulttiyrityksessä muutamille suomalaisille asiakkaille tehden pääasiassa asiantuntijakonsultointia erilaisissa ERP-projekteissa. Yksi asiakkaistani on lähtenyt tutustumaan Power Appsin mahdollisuuksiin ja haluaa tutkia voisiko sitä hyödyntää heidän teknologiaratkaisuissansa tulevaisuudessa.

Jatkossa yhä enemmän sovelluksia tehdään low-code-teknologioilla eli perinteisen ohjelmoinnin sijaan sovellukset tehdään drag & drop -tyylisesti graafisessa käyttöliittymässä. Näin toimii myös Microsoftin Power Apps, jota pidetään tällä hetkellä yhtenä varteenotettavimmista sovelluksista kyseisellä alueella. Gartner arvioi omassa Magic Quadrant -raportissaan, että vuoteen 2025 mennessä jopa 70 % uusista sovelluksista tehtäisiin low-code-menetelmillä. Heidän raportissaan nimenomaan Power Apps sijoitetaan johtavien sovellusten joukkoon ja sen arvioidaan saavuttavan suurimman käyttäjäkunnan johtuen laajalle levinneestä Office 365:n ja Microsoft 365 Dynamicsin käytöstä. (Wong, Iijima, Leow, Jain & Vincent 2021.)

Power Apps kuuluu Microsoftin Power Platformiin, joka on kattavien liiketoimintaratkaisujen rakentamiseen tarkoitettu kokonaisuus. Power Appsin lisäksi Microsoftin sivuilla Power Platformiin on kirjattu kuuluvan Power BI, Power Pages, Power Automate ja Power Virtual Agents. Microsoft kuvaa Power Platformin olevan enemmän kuin osiensa summa. (Microsoft s.a.) Tämän opinnäytetyön kontekstissa low-code-sovelluksista vahvasti esillä ovat Power Apps ja Power Automate unohtamatta Power BI:tä, jonka avulla data voidaan näyttää visuaalisessa muodossa korostaen liiketoiminnan kannalta tärkeitä tietosisältöjä.

Power Platform on käytössä sekä isoilla että pienillä yrityksillä ja se on tutkitusti auttanut liiketoimintoja saavuttamaan paremman ROI:n, virtaviivaistamaan liiketoiminta- ja kehitysprosesseja sekä isosti vähentämään sovelluskehitykseen palavia kuluja.

1.1 Työtehtävissä tarvittava osaaminen

Päivittäisissä työtehtävissäni vaaditaan paljon erilaista osaamista ICT:n eri osa-alueilta. ERP-järjestelmien asiantuntijana kohtaan päivittäin erilaisia haastavia tilanteita, joissa vaaditaan ongelmanratkaisukykyä, paineensietokykyä sekä luovaa ajattelua laatikon ulkopuolelta. Teen järjestelmiin liittyviä teknisiä toimenpiteitä, kuten datamigraatioita, testausta sekä konfigurointia. Olen mukana monenlaisissa kehityshankkeissa suunnittelemassa ja toteuttamassa erilaisia muokkauksia ja jatkuvan kehityksen tarpeita. Tuen järjestelmien pääkäyttäjiä, liiketoiminnan edustajia sekä loppukäyttäjiä heidän kohtaamisissa haasteissaan. SQL-kyselykielen hallinta on yksi tärkeimmistä työskäni tarvittavista teknisistä taidoista. Lisäksi tarvitsen sujuvaa kirjallista sekä suullista suomen ja englannin kielen taitoa.

Tämän opinnäytetyön kontekstissa tarvittava osaaminen linkittyy aiempaan laajaan ERP-osaamiseen, aiemmin hankittuun tekniseen osaamiseen kuten SQL:ään sekä ohjelmoinnin perusteisiin, asiakkaan liiketoiminnan ymmärtämiseen sekä moniin Microsoftin työkaluihin kuten Power Platform tuoteperheeseen kuuluvaan Power Appsiin, Power Automateen sekä Power BI:hin, Active Directoryyn, Azureen ja Office 365:een.

1.2 Ammatillisen kehittymisen tavoitteet ja rajaukset

Kaksi kokonaisuutta, joiden avulla tässä opinnäytetyössä osaamiseni kehittämistä seuraan ovat asiakkaan Power Apps kehitysprojekteja. Toisessa POC-hankkeessa pyrimme kehittämään sovellusta, jonka olisi tarkoitus korvata asiakkaan tuotantokäytössä oleva ERP-järjestelmän oma mobiilisovellus. Nykyisellään mobiilisovellusta käytetään työ- ja päivystysajan leimaamiseen, työtilausten käsittelyyn, ostojen ja kulujen kirjaamiseen sekä kaluston raportointiin. POC-hankkeen tavoitteena on testata Power Appsin kyvykkyyksiä sovelluksen korvaamisessa. Tarve POC-hankkeen käynnistämiseksi tulee asiakkaan liiketoiminnoilta, jotka ovat kyllästyneitä ERP-järjestelmään kuuluvan mobiilisovelluksen kehittämisen mahdottomuuteen sekä siinä alati esiintyviin ongelmiin kuten synkronointi-, alustus- ja päivitysongelmiin.

Toisessa jo pidemmälle edenneessä projektissa luomme HR-osastolle järjestelmää, jonka avulla HR voi seurata henkilöstön tekemiä häirintäilmoituksia, kirjata ilmoitusten selvitykseen liittyviä lisätietoja sekä analysoida datan perusteella mahdollisia ajankohtia, jolloin ilmoitusten määrä on kasvussa. Tässä projektissa ehdottoman tärkeässä roolissa ovat käyttöoikeudet, sillä arkaluontoisen datan vuoksi on varmistuttava, ettei ylimääräisillä henkilöillä ole oikeuksia nähdä ilmoitusten sisältöä. Lisäksi niiden käyttäjien osalta, joilla on oikeus nähdä ilmoituksia, käyttöoikeuksissa on paljon rajoituksia käyttäjän roolin mukaan, joten se tulee huomioida myös käyttöoikeuksia suunnitellessa ja luodessa.

Taulukko 1. Peittomatriisi päiväkirjaopinnäytetyön tekstinsisäisistä kytköksistä

Oman ammatillisen kehittymisen tavoitteet	Seurantaviikko	Oman ammatillisen kehittymisen tulokset
Teknisen osaamisen kasvattaminen low-code-tekniologioissa	1, 2, 3, 4, 6, 7, 8	3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8
Asiakkaan kannalta oleellisten asioiden tunnistaminen ja priorisoitu edistäminen	1, 2, 4, 5, 6, 8	3.1, 3.2, 3.4, 3.5, 3.6, 3.8
Ketterän ohjelmistokehityksen (Agilen) teorian ja käytännön ymmärtäminen low-code-viitekehityksessä	1, 2	3.1, 3.2

Peittomatriisin tavoitteena on havainnollistaa tavoitteen ja seurantaviikon sekä kehittymisen tulosten välistä yhteyttä.

1.3 Tietoperusta

Aion hyödyntää lähdemateriaalina mahdollisimman ajantasaista dokumentaatiota ja tietoa kuten Power Appsin Microsoft MVP kehittäjien blogikirjoituksia, YouTube-videoita sekä muun muassa Microsoft Learn oppimisympäristöä, josta löytyy Power Appsiin erittäin ajantasaista dokumentaatiota sekä erilaisia oppimispolkuja. Näistä esimerkkinä Power Apps funktionaalisen konsultin sertifikaattiin tähtäävä kurssi PL-200 ja yleisesti Power Platformin fundamentaalista ymmärrystä kasvattava PL-900.

Power Apps kehittyä jatkuvasti Microsoftin julkaistessa uusia päivityksiä, joten vuoden tai parin vanha teos Power Appsisista alkaa olla jo painosta tullessaan vanhentunut muun muassa ominaisuuksien, mahdollisuuksien ja teknisten rajoitusten näkökulmasta. Sen sijaan esimerkiksi Microsoft Learnin oppimispolkujen materiaaleja ja Power Appsin omaa dokumentaatiota päivitetään jatkuvasti vastaamaan nykyhetken Power Appsia ja sen tarjoamia mahdollisuuksia.

1.4 Keskeiset käsitteet

Tässä luvussa avaan opinnäytetyön ymmärtämisen kannalta tärkeimpiä käsitteitä kahteen eri kategoriaan jaoteltuna.

1.4.1 ICT-alan ammattikäsitteet ja asiakkaan käyttämä terminologia

ALM (Application lifecycle management) – Sovelluksen elinkaarta kuvaava termi, joka pitää sisällään sovelluksen hallinnoinnin, kehityksen ja ylläpidon (Microsoft 2022f).

BAS (Business Application Services) – Asiakkaan kumppaniyrityksen toiminto, joka pitää huolta asiakkaan liiketoimintakriittisistä sovelluksista ylläpidollisesti, tukee ongelmatilanteissa sekä osallistuu kehityshankkeisiin.

Debuggaus – Ohjelmistokehityksen vaihe, jossa virhetilanteen aiheuttaja pyritään löytämään ja ratkaisemaan (Heusser 2019).

Demo – Esitys, jonka avulla voi näyttää tuotteen tai palvelun toimivuutta. Teknisellä demolla voidaan demota esimerkiksi sovelluksen toimintoja. Myyntidemo on suunnattu myynti-ihmisille ja johtajille. (Hart 2021.)

DS (Dynamic Scheduling) – Asiakkaan käyttämään ERP-järjestelmään linkittyvä optimointityökalu, joka laskee erilaisten parametrien perusteella työtilauksille sopivimman tekijän.

ERP-järjestelmä – Toiminnanohjausjärjestelmä, johon voi kuulua muun muassa kirjanpito, laskutus, tuotannon suunnittelu, henkilöstön hallinta, kaluston hallinta, asiakas-, kohde- ja sopimustietojen hallinta sekä paljon muuta. Useimmiten räätälöitävissä yrityksen tarpeiden mukaan.

IAL (Information Access Layer) – Asiakkaan ERP-järjestelmän mukainen rajapinta, josta integraatio hakee tietoa ja siirtää sitä Dataversen puolelle. IAL on käytännössä näkymä tai taulu, johon on tuotu tarvittavat tiedot kätevämmiin saataville muista asiakkaan ERP-järjestelmän näkymistä.

Integraatio – Kahden tai useamman järjestelmän välinen palvelu, joka siirtää tietoa järjestelmien välillä. Integraatioita voi olla sekä yksisuuntaisena, jolloin lähettävä järjestelmä lähettää ainoastaan vastaanottavaan järjestelmään tai kaksisuuntaisia, jolloin tiedot liikkuvat molempiin suuntiin. (Valjas 2019.)

Integraatiotäppä – Asiakkaan sopimusrivillä oleva valintaruutu, jonka avulla integraatio pystyy päättämään, että kuuluuko kohde siirtää integraatiossa vai ei.

JSON (Javascript Object Notation) – Tehokas ja moderni datan tiedonsiirtomuoto, jota käytetään tietojen välittämiseen järjestelmien välillä. Helposti sekä ihmisen että koneen käytettävissä (json.org s.a).

Keikka – Asiakkaan käyttämä termi, joka kuvaa sopimukseen kuulumattomien töiden tekemistä.

Ketterä kehitys (Agile Software Development) – Kuvaa joustavaa ohjelmistokehitystä, jossa tärkeässä roolissa ovat henkilöt, kanssakäymiset asiakkaan kanssa sekä toimiva ohjelma. Ketterän kehityksen avulla tiimit voivat vastata nopeasti muuttuviin tilanteisiin. (Drumond s.a.)

Low-code – Sovelluskehitystä, jossa käytetään ensisijaisesti graafista käyttöliittymää sovelluksen kehittämiseen. Low-code-alustojen avulla sekä kehittäjät, että ei-kehittäjät voivat nopeasti luoda, jakaa ja muokata sovelluksia yritysten ja organisaatioiden liiketoimintatarpeisiin.

mvp (minium viable product) – Tuote tai palvelu, jossa on vähimmäismäärä ominaisuuksia, jotta sitä voidaan testata ja sitä kautta kerätä palautetta tulevaisuuden kehittämistä varten (Shufeldt 2021). Suomeksi pienin toimivin tuote.

Mäppäys – Prosessi, jonka avulla lähde- ja kohdedata saadaan kommunikoimaan keskenään. Mäppäystä voidaan käyttää esimerkiksi integraatioissa, migraatioissa ja datan muuntamisessa. (Keating 2021.)

No-code – Pohjimmiltaan samanlainen low-coden kanssa mutta nimensä mukaisesti kehittäjä ei luo koodia ollenkaan. No-codella voi tällä hetkellä tehdä hyvin yksinkertaisia sovelluksia. No-code ei sovellu monimutkaisempiin ratkaisuihin.

POC (Proof of concept) – Virallisesti termillä viitataan testaamiseen, jossa kartoitetaan voisiko idean muuttaa todeksi mutta käytetään usein laajemmassa kontekstissa (Malsam 2021). Opinnäytetyön kontekstissa POCilla kuvataan esimerkiksi hanketta, joka on jo aloitettu, ja josta on jo toimiva esitettävissä oleva versio olemassa.

Pääjärjestelmä – Asiakkaan termi ERP-järjestelmän tietokoneella käytettävälle versiolle.

Rajapinta – Rajapinta mahdollistaa sovelluksen tietojen lukemisen tai sovellukseen tietojen lähettämisen. Rajapinta mahdollistaa integraation sovellukseen tai järjestelmään mutta käyttö vaatii käyttöoikeuksia. (Valjas 2019.)

Relaatiotietokanta – Koostuu useista tauluista, joiden data on riveissä ja sarakkeissa. Taulujen välillä on yhteyksiä. Yhdistettävissä tauluissa täytyy olla sama avain eli ID-kenttä. Toisessa taulussa avain on pääavain (primary key) ja toisessa viiteavain (foreign key). (Oracle s.a.)

Responsiivisuus – Tässä yhteydessä tarkoitetaan sovelluksen responsiivisuutta eli sovelluksen kykyä pysyä käytettävänä, vaikka näytön koko vaihtuisi mobiililaitteen, tabletin tai ison tietokone-näytön välillä. Sovelluksen osat muuttuvat automaattisesti näytön koon muuttuessa pitäen sovelluksen koko ajan käyttökelpoisena.

Scrum – On ketterän kehityksen viitekehys, jonka avulla pyritään tuottamaan arvoa ratkaisemalla kompleksisia ongelmia joustavasti. Scrum pitää sisällään kehitysjonon, sprintin eli aikavälin, jolloin kehitystä tehdään inkrementaalisesti sekä tulosten tarkastelun (retrospektiivi). Scrumiin kuuluu päivittäispalaverit (Daily Scrum), joissa käydään kuhunkin sprinttiin kuuluvia tehtäviä Scrum Masterin johdolla läpi. (Schwaber & Sutherland 2020, 3–12.)

SOAP (Simple Object Access Protocol) – Sovellusten välinen protokolla, jonka mahdollistaa kommunikaation XML:n avulla (W3schools s.a.).

SQL (Structured Query Language) – Kyselykieli, jonka avulla relaatiotietokannoista voidaan hakea tietoa. SQL:n avulla voidaan myös luoda, muokata ja poistaa tietoja.

Tietokanta – Pitää sisällään järjestettyä tietoa, useimmiten taulukoissa, joissa on rivejä ja sarakkeita. Tämän opinnäytetyön kontekstissa tietokannat ovat relaatiotietokantoja.

XML (Extensible Markup Language) – On merkintäkieli, jota käytetään tiedonvälitykseen järjestelmien välillä sekä dokumenttien tallentamiseen (Microsoft 2022i).

1.4.2 Microsoftin tuotteisiin ja palveluihin liittyvät käsitteet

Azure – Microsoftin julkinen pilvipalvelu tai tarkemmin ehkä alustapalvelu, joka skaalautuu monenlaisten yritysten ja organisaatioiden tarpeisiin. Azure kattaa kokoelman integroitua palveluja, joita voi hyödyntää sovellusten kehittämiseen, suorittamiseen ja hallinnoimiseen. (Bergius 2014.)

Azure Data Factory (ADF) – Azuren palvelinvapaa integraatiokeskus, johon voi integroitua yli 90 liittimen (connector) avulla. ADF:llä pystyy myös luomaan ELT ja ETL prosessit joko koodaamalla tai ilman - hyödyntämällä graafista käyttöliittymää. (Microsoft 2022j.)

Canvas-sovellus – Power Appsin toinen mahdollinen sovellussuunnittelun lähestymistapa Model driven-sovelluksen lisäksi. Canvas-sovelluksessa nimensä mukaisesti kehitys aloitetaan tyhjästä ja Canvas-sovellukset tarjoavat Model Driven-sovelluksiin nähden paljon enemmän kustomointimahdollisuuksia.

Kokoelma (collection) – On Power Appsin sisällä oleva ohjausobjekti, jonka sisään voi tallentaa tietoa ja jota voidaan hyödyntää tallennettujen tietojen etsimiseen.

Microsoft Dataverse – Tietokanta, johon muun muassa Power Appsin tiedot voidaan tallentaa.

Microsoft MVP (Most valuable professional) – Microsoftin vuosittain jakama palkinto ahkerasta yhteisöllisestä panoksesta jonkin tietyn osa-alueen hallitsemisessa (Microsoft s.a.).

Patch – Power Appsin funktio, jonka avulla tietoa saadaan luotua tai muokattua tietolähteeseen kuten Microsoft Dataverseen tai Microsoft SharePointiin.

Power Apps – Microsoftin Power Platform tuoteperheeseen kuuluva sovellus, jonka avulla pystyy tekemään sovelluksia low-code-periaatteilla ja integroimaan niitä saumattomasti Microsoftin sekä kolmansien osapuolien palveluihin.

Power Automate – Sovellus, jonka avulla tehtäviä voidaan automatisoida. On hyvin integroitavissa sekä Microsoftin että kolmansien osapuolten palveluiden ja tuotteiden kanssa.

Power BI – Microsoftin raportointityökalu, jonka avulla voidaan luoda muun muassa kattavia interaktiivisia raportteja.

Power Platform – Microsoftin tuoteperhe, johon muun muassa Power Apps, Power Automate ja Power BI kuuluvat.

Säilö (container) – Ohjausobjekti, jonka sisällä voi olla monenlaisia muita ohjausobjekteja kuten tekstikenttiä (label), tekstinsyöttökenttiä (input), painikkeita (button) ja pudotusvalikoita (dropdown).

Triggeri / liipaisin – Power Automatessa käytetty termi, joka kuvaa Power Automaten ensimmäistä vaihetta, joka käynnistää työnkulun suorituksen.

Työnkulku – Kuvaa Power Automatella tehtäviä työnkuluja eli vaiheittaisia toimenpiteitä, joiden tavoitteena on automatisoida usein toistettavia tehtäviä ja helpottaa sovelluksen toiminnallisuuksien luomista.

2 Lähtötilanteen kuvaus

Teen lähtötilanteen kuvauksen vain opinnäytetyön kontekstissa ja jätän siitä kaikki muut nykyiseen työnkuvaani kuuluvat työtehtävät pois. Tämän tavoitteena on selkeyttää opinnäytetyön rajausta ja mahdollistaa parempi pohdinta lähtötilanteen ja seurantajakson välillä tapahtuneesta osaamisen kehittämisestä.

2.1 Oman nykyisen työn analysointi

Olen aloittanut Power Appsin ja siihen liittyvien muiden tuotteiden harjoittelun viime vuoden puolella, kun asiakas alkoi tutustumaan Power Appsin mahdollisuuksiin ja allokoit minut kehitysprojektiin mukaan. Osaamista olen kerryttänyt tutustumalla etenkin Microsoft MVP henkilöiden julkaisemiin YouTube-videoihin, artikkeleihin ja blogikirjoituksiin. Lisäksi olen tutkinut Microsoftin omaa kattavaa dokumentaatiota ja hyödyntänyt yksittäisiin asioihin täsmällisiä google-hakuja. Olen myös hieman tutustunut Microsoft Learnin tarjoamaan koulutusmateriaaliin Power Appsiin liittyen.

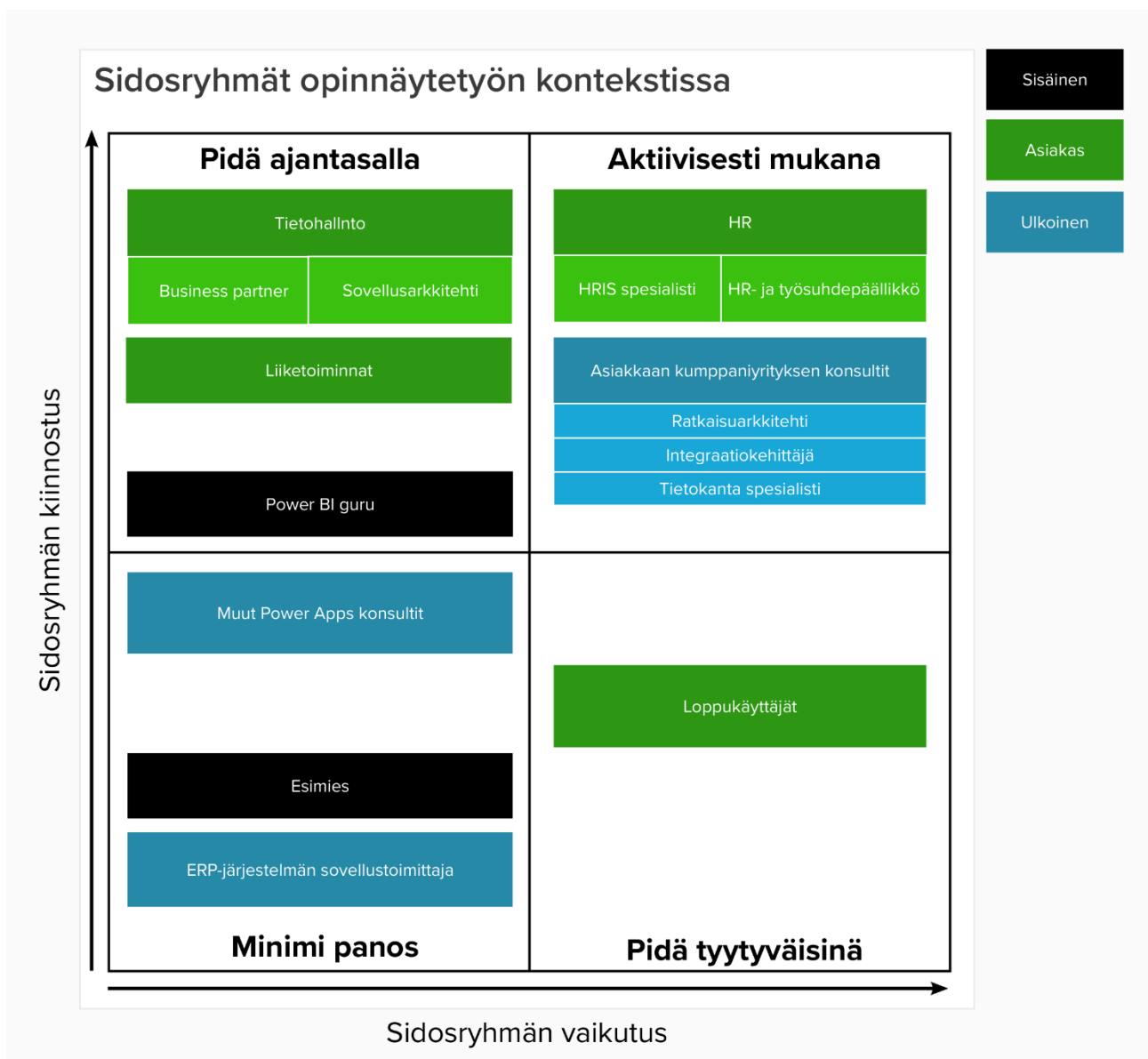
Konkreettisia työtehtäviä ovat muun muassa käyttöliittymän suunnittelu ja toteuttaminen. Integraatioiden suunnittelu, toteuttaminen ja testaus yhdessä asiakkaan kumppaniyrityksen ratkaisuarkkitehdin, integraatiokehittäjän sekä tietokantaspecialistin kanssa. Edellä mainituista tehtävistä selviämiseen tarvitaan Power Apps-kehitysympäristön, Dataverse-tietokannan, Power Automaten sekä Azuren eri tuotteiden ymmärrystä. Lisäksi asiakkaan nykyisen ERP-järjestelmän tekninen ja käytännöllinen tuntemus tuo arvokasta apua integraatiokehitykseen. Ohjelmoinnin perusteiden ymmärtäminen auttaa hahmottamaan tarvittavien funktioiden ja toiminnallisuuksien luomisessa.

Arvioisin itseäni Power Platformiin liittyen aloittelevaksi toimijaksi, sillä tarvitsen edelleen paljon taustamateriaalia, yritystä ja erehdystä sekä välillä myös konsultaatioapua kokeneemmilta Power Platform osaajilta. Soveltamalla aiemmin mainitsemiani materiaaleja sekä kokeilemalla erilaisia asioita, olen jo kuitenkin pystynyt luomaan asiakkaan hyväksymiä ratkaisuja ja pysynyt tehtävieni tasalla.

2.2 Sidosryhmien esittely

Esittelen tärkeimmät sidosryhmät ja sidosryhmien intressit valta-kiinnostus-matriisin (Indeed Editorial team 2021) avulla alla olevassa kuvassa 1. Sidosryhmät jakautuvat kolmeen eri ryhmään. Sisäiset sidosryhmät ovat oman yritykseni sidosryhmiä, joiden intressit ja osallistuminen ei ole kovinkaan suurta päivittäisessä tekemisessäni. Asiakkaan sidosryhmät sen sijaan jakautuvat jonkin verran eri kategorioihin ja esimerkiksi HR:ään liittyvät sidosryhmät ovat aktiivisesti mukana päivittäisessä tekemisessä. Tietohallinto ja liiketoiminnot on hyvä pitää ajan tasalla, ja heidän kanssaan

olen tekemisissä vähintään viikoittain. Loppukäyttäjien intressit tulevat esiin myöhemmin, kun kehitykset ovat siinä vaiheessa, että niitä aletaan ottamaan käyttöön. Teen paljon yhteistyötä päivittäin myös ulkoisten sidosryhmien kuten eri alueen asiantuntijakonsulttien kanssa.



Kuva 1. Sidosryhmät

2.3 Työpaikan vuorovaikutustilanteet

Lähes kaikki työssäni olevat vuorovaikutustilanteet tapahtuvat nykyään virtuaalisesti, sillä olen työskennellyt jo kohta kolmen vuoden ajan täysipäiväisesti kotoa käsin. Viralliset asiat hoituvat Outlook-sähköpostin sekä tikettijärjestelmien kautta. Suurin osa kommunikaatiosta tapahtuu kuitenkin pikaviestien ja palaverien välityksellä Teams-sovelluksen avulla. Joitakin palavereja, oman tiimin tapaamisia tai asiakkaan tapaamisia saattaa olla myös paikan päällä.

Olen huomannut, että oikea-aikainen puheen aloitus palaverissa tuottaa välillä haasteita ja siinä minulla olisi parannettavaa. Saatan joskus vahingossa puhua toisen päälle keskeyttämällä toisen henkilön aloittaman jutun. Siitä syntyy usein muutaman sekunnin kestävä kiusallinen hetki, jonka jälkeen jompikumpi jatkaa omaa asiaansa. Pärjään englannin kielen kanssa erinomaisesti palaverissa sekä kirjallisessa viestinnässä kuten sähköposteissa ja Teams-viesteissä. Puhumiseen kaipaisin kuitenkin vielä lisää sujuvuutta, jotta keskustelu soljuisi entistä paremmin. Tätä pystyn harjoittelemaan parhaiten puhumalla mahdollisimman paljon englantia eri ihmisten kanssa.

Havaintoni mukaan osa vielä käyttää Outlook-sähköpostia asioiden hoitamiseen, jotka hoituisivat mielestäni paremmin esimerkiksi Teamsilla. Pysin viemään kaiken kommunikoinnin juuri Teamsiin, sillä sen avulla asiat hoituvat mielestäni kaikista joustavimmin ja sieltä on myös helpompi löytää aiemmin käydyt keskustelut ja mahdollisesti niissä jaetut asiat. Nykyajan vuorovaikutuksessa on tärkeä pystyä hyödyntämään reaaliaikaisia työkaluja kuten Teamsia ja Jiraa, jotta tilanteisiin ei tule liikaa viivästystä ja mahdollisimman paljon dokumentaatiota ja tietoa löytyisi yhden väylän kautta eikä sitä tarvitsisi etsiä esimerkiksi lähetettyjen tai saapuneiden sähköpostien joukosta.

Olen todennut monissa tapauksissa ryhmäkeskustelun muodostuksesta olevan apua. Sitä voidaan käyttää esimerkiksi käyttöönottilanteissa kuumana linjana, jonka kautta saa helposti kollegat ja muut tarvittavat tahot kiinni. Testauksessa sitä voidaan hyödyntää kysymysten esittämiseen ja jälleen kerran ihmisten kiinni saamiseen. Yritän nykyään aina sopivassa tilanteessa ehdottaa ryhmäkeskustelun muodostamista sen sijaan, että asioita hoidettaisiin sähköpostin välityksellä. Olen varpaillaan etenkin silloin, jos huomaan, että asiasta on käyty pitkiä sähköpostikeskusteluja eikä asia lähde ratkeamaan.

3 Seurantajakson raportointi viikkoanalyysineen

Tässä luvussa esitetään työviikon päiväkirjamerkinnot, mikäli kyseisinä päivinä on ollut opinnäytetyön kontekstiin kuuluvia työtehtäviä. Merkinnoissa kuvataan päivittäin tavoitteet, pyritään arvioimaan niiden toteutumista sekä jokaisen viikon lopussa analysoidaan viikon tapahtumat ja vedetään osaamisen kehittäminen yhteen viikkoanalyysin muodossa.

3.1 Seurantaviikko 1 (vko 35)

Maanantai 29.8.2022

Tavoite: Selvittää loppuun viime viikolla alkaneen työtilauksen käsittelyn testauksessa nousseen integraatio-ongelman syy. Saimme jo osittain korjattua ongelmaa korjaamalla integraation käyttämän sanoman lyhyempään muotoon, jonka myös testasin sovelluksen omalla palvelintesti-työkälyllä. Nyt integraatioajo jää kuitenkin Azuressa kokonaan jumiin ja tavoitteena on ratkaista miksi näin käy. Integraatio-ongelman selvityksessä kehittyvät niin integraatio kehittämisen ajattelu, teknisen toteutuksen ymmärrys Azuressa kuin virheen tulkinnan ymmärrys.

Toteutuminen: Joka viikkoisessa ICT tilanne check -palaverissa sivusimme lopussa POC-hankkeessa tullutta ongelmatilannetta ja sovimme, että miten selvityksen kanssa edetään. Sain palaverin jälkeen kuulla ratkaisuarkkitehdiltä, että integraatiokehittäjä on poissa töistä sairastumisen vuoksi emmekä sen vuoksi pysty jatkamaan ongelman selvitystä taikka muitakaan POC-hankkeen tärkeimpiä integraatiokehityksiä.

Vastaan tulleiden haasteiden vuoksi käytin aikaa käyttöliittymän paranteluun muun muassa korjaamalla työtilaukset-näkymän työtilauskortteja sekä sovelluksen lomakkeita visuaalisesta näkökulmasta. Huomioin myös alusta asti pitämäni responsiivisen suunnittelun ajattelun muutoksia tehdesäni. Responsiivisella suunnittelulla tarkoitetaan sovelluksen mukautumista käyttäjän näyttökoon mukaan. Responsiivisuus Power Appsissa saadaan aikaiseksi säilöillä (container) sekä eri ohjausobjektien leveys, korkeus, x ja y parametrien avulla. Käyttöliittymän visuaalinen osuus on tässä vaiheessa POC-hanketta vielä pienellä prioriteetilla, joten tähän täytyy palata myöhemmin, kun muut tärkeämmät asiat on saatu kehitettyä.

Tiistai 30.8.2022

Tavoite: Jatkaa eilisen ongelman selvitystä ja löytää siihen ratkaisu.

Toteutuminen: Totesimme keskustellessamme ratkaisuarkkitehdin kanssa, että olisi hyvä saada Azuren päähän oikeudet useammalle henkilölle heidän päässään, jottei kehitys tyssäisi jatkossa yhden henkilön poissaoloon. Sovimme, että hoidan asiaa asiakkaan ICT:n palvelujen toimituksesta

vastaavan henkilön kanssa, sillä minun piti muutenkin olla häneen yhteydessä Power Apps ympäristössä havaitsemani käyttöoikeushaasteen vuoksi. Power Appsissa tai asiakkaan ympäristössä oli tapahtunut jokin muutos, joka esti minua muun muassa muokkaamasta ja luomasta Dataverse-tiloja. Löysin virheestä lisätietoa googlaamalla ja ymmärsin, että minulta puuttuu ympäristöstä järjestelmänvalvojan oikeudet.

Ratkaisuarkkitehdiltä sain linkin ja tiedot tarvittavista oikeuksista muutamalle muulle henkilölle heidän puolellansa. Testasin myös linkkiä ja huomasin, että minulla oli jo valmiiksi pääsy tarvittaviin paikkoihin. Tutkin epäonnistunutta ajotapahtumaa, sen logiikkaa ja virheilmoitusta tarkemmin.

```
{
  "message": "The character set provided in ContentType is invalid. Cannot read content as string using an invalid character set.",
  "status": 500,
  "error": {
    "message": "The character set provided in ContentType is invalid. Cannot read content as string using an invalid character set."
  }
}
```

Virheilmoituksen viestin "ContentType" -kohta herätti huomioni, vei oikeille jäljille ja sen avulla löysin Azuren integraatiologiikasta virheen, jonka vuoksi muutossanoma jäi virheeseen. Ilmoitin havainnot yhteisessä Teams-ryhmäkeskustelussamme sairaslomalta palanneelle integraatiokehittäjälle ja hän teki tarvittavat korjaukset Azuren puolelle.

Testasimme korjausten jälkeen integraatiota ja se toimi odotetusti eli virheet hävisivät ja ajot menivät läpi onnistuneesti.

En ollut aiemmin koskenut Azuren päähän millään tavalla, mutta tänään huomasin, että se on todella samanlainen, kun Power Automate. Tutkiessani aiemmin mainittua virhettä, tulin myös paremmin tietoiseksi kyseisen integraation logiikasta. Olin aiemmin nähnyt vain pieniä palasia sieltä täältä integraatiokehittäjän näytön jakojen kautta.

Keskiviikko 31.8.2022

Tavoite: Tavoitteena luoda testitapauksia, joilla voimme testata aiemmin korjatun työtilausintegraation toimivuutta liiketoiminnan käyttötapausten pohjalta. Lisäksi tavoitteena on tutkia mikä olisi sopiva tapa tehdä muita toimenpiteitä kuten työaikakirjausten luomista Power Appsista asiakkaan ERP-järjestelmään. Tähän sopivaa rajapintaa ei ole valmiiksi kehitettynä ja tavoitteena varmistaa voidaanko suorita SQL-suorituksia hyödyntää tässä tapauksessa. Päivän tavoitteet liittyvät vahvasti teknisen ymmärryksen kasvattamiseen, sillä muun muassa tietokantasuorituksiin tulen tarvitsemaan tietokantaspecialistin tukea. Ymmärrän niistä jonkin verran ja osaisin varmasti itsekin niitä

tehdä, mutten ole kuitenkaan asiakkaan ympäristössä niitä käytännössä aiemmin tehnyt ja tiedän, ettei tietokantaspesialistilla niissä kauan mene.

Toteutuminen: Meillä on käytössä Jira, johon olemme kirjanneet hankkeen ylätasoin suunnitelmaa sekä hankkeen tehtäviä kanban-tauluun. Kirjasimme Jiraan uuden tehtävän testauksesta ja pohdimme liiketoiminnan tarpeita työtilauksen käsittelyssä ja pyrimme kirjaamaan erilaisia käyttötappauksia työtilauksen käsittelystä. Testauksessa totesimme, että perustapaus eli työn hyväksyntä, aloitus ja valmiiksi kuittaaminen toimi oikein. Tämän lisäksi saimme myös työn keskeytyksen toteutettua, joskin hieman workaround-tyyppisellä menetelmällä. Siinä keskeytys saatiin aikaiseksi hyödyntämällä sellaista tilaa, jota ei ole rajapintaan määritelty mutta joka kuitenkin tuottaa halutun lopputuloksen. Tähän täytynee myöhemmin pohtia parempi ratkaisu, sillä workaround-ratkaisu ei nimensä mukaisesti ole kovinkaan pitkäaikainen eikä siisti ratkaisu.

Torstai 1.9.2022

Tavoite: Tavoitteena on testata eilen työtilauksen käsittelyssä ilmenneiden epäonnistuneiden tilanmuutosten vaihtoehtoista toteutustapaa yhdessä ratkaisuarkkitehdin ja tietokantaspesialistin kanssa.

Toteutuminen: Pidimme varatun testauspalaverin suunnitellusti ja testasimme suoraan asiakkaan testiympäristön tietokannassa PL/SQL-päivityksen käyttämistä SOAP-rajapinnan kautta testattujen epäonnistuneiden tilanmuutosten tekemiseksi. Meillä oli kolme erilaista käyttötappautta testissä, joista kaksi vaikuttivat selvemmillä ja kolmas hieman monimutkaisemmalta. Hyödynsimme ERP-järjestelmän debuggerityökalua, jonka avulla saimme käyttöliittymässä toimenpiteitä vastaavat tietokantametodit, joiden avulla pystyimme testaamaan eri käyttötappauten mukaisia tilanmuutoksia.

Perjantai 2.9.2022

Tavoite: Suurin osa viikosta on mennyt POC-hankkeen edistämiseksi mutta HR-projektia täytyy myös pystyä edistämään, sillä sen sovittu tuotantovalmius (1.10.) lähestyy. Keskenäisiä asioita ovat muun muassa raportointi sekä Microsoft Formsin kautta tulevien ilmoitusten logiikan muutokset Power Automateen. Jälleen teema viettää vahvasti teknisen puolelle ja tänään todennäköisesti tulen oppimaan lisää Power Automatesta, siinä kehityksen aikana ilmenevistä virheistä ja niiden debuggaamisesta.

Toteutuminen: Kävimme kollegani, Power BI gurun kanssa läpi HR-sovelluksen raportoinnin toteutusta Power BI:n puolella. Hänelle oli hieman epäselvää, kuinka käyttöoikeudet rajaavat näkyvyyttä ja miten niitä voidaan hyödyntää Power BI:n puolella. Avasin käyttöoikeuksien logiikkaa, jonka jälkeen kollegani ymmärsi asian ja ryhtyi rajaamaan raporttia tarvittavien muutosten avulla.

Tein aiemmin sovittuja muutoksia Power Automaten tapaan käsitellä Microsoft Formsilla lähetettyjä ilmoituksia. Power Automaten muutosten testaukseen lähes kuuluu virheelliset ajot sekä niiden debuggaus ja korjaus. Virheitä tuli jonkin verran ja jonkin verran meni myös aikaa niiden selvittämiseen. Lopulta sain Power Automaten työnkulun toimimaan halutulla tavalla ja Microsoft Formsista lähetetyt ilmoitukset määpötyivät Power Automaten kautta Power Apps sovellukseen oikein.

Viikkoanalyysi 1

Viikon päätavoitteena oli saada POC-hankkeen työtilausintegraation ongelmat selvitettyä ja integraatio toimimaan sulavasti Power Appsin ja asiakkaan ERP-järjestelmän välillä. Integraatiokehittäjän viime viikon logiikkavirheen vuoksi alkuvuokosta aikaa meni hukkaan ongelman selvitykseen, jossa ajo jäi Azuresa virheeseen. Minulle jäi viime viikolla käydyn ryhmäkeskustelun viestien perusteella hieman epäselväksi mikä varsinainen ongelma oli ja miten sitä oli yritetty ratkoa. Lisäksi integraatiokehittäjän sairastuminen viivästytti kehitystä. Tästä oppina jäi käteen, että ketterän kehityksen kannalta olisi hyvä olla useampi henkilö, jolta löytyy riittävät oikeudet kehityksen edistämiseen ja vastaan tulevat ongelmat olisi hyvä käydä heti yhdessä läpi, jotta kaikilla on selvä kuva ongelman laadusta, vakavuudesta sekä sen mahdollisista ratkaisuvaihtoehdoista.

Ei ole aina selvää missä vaiheessa olisi sopiva hetki ottaa palaveri ja missä vaiheessa asioiden hoitaminen kirjallisesti toimii paremmin. Shani Harmonin (2019) mainitseminen EPIC (emotional, purposeful, interpersonal, complexity) yhteistyön sääntöjen mukaan kompleksiset ongelmat kannattaa hoitaa palaverin avulla, sillä sitä kautta on epätodennäköisempää, että tärkeät yksityiskohdat jäävät huomaamatta. Epäilisin, että tässä tapauksessa kävi juuri niin, että tärkeä yksityiskohta jäi integraatiokehittäjältä huomaamatta ja hän pohti jotakin erilaista, mahdollisesti monimutkaisempaa ratkaisua, vaikka loppupeleissä ratkaisu olikin hyvin yksinkertainen.

Ratkaisu ongelmaan löytyi tutkimalla Logic Appsin työnkulun epäonnistuneen tapahtuman eri vaiheita. Tein tämän aluksi hyvin intuitiivisesti, vuosien saatossa kehittyneen ongelmanselvitys ajattelutavan avulla. Lähdin liikkeelle tärkeimmästä eli virheilmoituksesta, joka usein vähintäänkin ohjaa oikeaan suuntaan ongelman ratkaisemiseksi (Cocca 2022). Tämän jälkeen tutkin tarkemmin mitä kyseinen työnkulku tekee ennen virheen ilmestymistä, tutkin työnkulun inputit ja outputit sekä eri ehdot ymmärtääkseni paremmin työnkulun eri vaiheita. Nämä ovat nimenomaan Microsoftin oman dokumentaation mukaisia vaiheita epäonnistuneen työnkulun selvityksessä. (Microsoft 2022a.) Opin tutkiessani Azure Logic Appsin toimintalogiikasta ja huomasin paljon samankaltaisuuksia Power Automaten kanssa. Molemmat ovat palveluita, jotka voidaan integroida Microsoftin ja kolmansien osapuolten palveluihin ja joiden tavoitteena on automatisoida prosesseja. Selvin ero näiden kahden välillä on kohderyhmä ja käyttötarkoitus. Azure Logic Apps on tarkoitettu haastavimpiin ja monimutkaisiin kehityksiin, jonka tekevät ICT-ammattilaiset, kehittäjät ja järjestelmän ylläpitäjät,

kun taas Power Automatea voivat käyttää myös loppukäyttäjät sekä muut Office 365 ympäristössä toimivat Power Apps luojat. (Microsoft 2022b.)

Yksi selkeä asia, jonka olen huomannut helpottavan virheiden debuggaamista on kehityksen tekeminen pienissä palasissa. Tätä olemme noudattaneet sekä POC-hankkeen että HR-projektin kehittämisen yhteydessä. Vähän kerrallaan kehittäminen ja jatkuva uusien asioiden testaaminen pitää palaset hyvin kasassa ja uusien virheiden juurisyyt löytäminen on helpompaa. (Neumann 2014.) Olemme myös alusta asti pyrkineet huomioimaan, että kehitämme asiakkaan kannalta oleellisia asioita, joiden avulla heidän on helpompi arvioida POCin kautta Power Apps mahdollisuuksia toimia korvaajana heidän nykyisen mobiilisovelluksensa tilalla. Tämä ajatustapa osuu yksiin myös yhden ITIL 4:seen kuuluvan peruseräteen kanssa, jonka mukaan asiat kannattaa pitää yksinkertaisina, käytännöllisinä ja kehittää vain sellaista, jota asiakas tarvitsee (Mathenge & Stevens-Hall 2019). Sen sijaan, että kehittäisin visuaalisesti miellyttävän käyttöliittymän, pyrin keskittymään aluksi mahdollisimman paljon sovelluksen toiminnallisuuksien sujuvuuteen.

Huomasimme testausten myötä, ettei alkuperäinen ajatus SOAP-rajapinnan käyttämisestä toiminutkaan aivan kaikissa tapauksissa. Käyttämämme rajapinta oli suunniteltu eri käyttötarkoitusta varten eikä siihen sen vuoksi ollut rakennettu mahdollisuutta kaikkiin työtilauksen tilanmuutoksiin. Tehokkaan integraation rakentamisessa tulee huomioida muun muassa joustavuus, tietoturva ja yksinkertaisuus. Mitä yksinkertaisemmin integraation pystyy toteuttamaan, sen tehokkaampi se on. Aina tehokkaan integraation tekeminen ei kuitenkaan suoraan ole mahdollista vanhan järjestelmän rajoitteiden vuoksi. (Nduati 2021.) Valitettavasti SOAP-rajapinnan kehittäminen tai kokonaan uuden SOAP-rajapinnan luominen ei ole mahdollista omin voimin, vaan siihen tarvitaan sovellustoimittajan tekemistä. Aika- ja kustannusrajoitteiden vuoksi totesimme, ettei sitä kannata tässä vaiheessa lähteä edistämään, vaan täytyy kehitellä jokin toinen ratkaisu näiden työtilausmuutosten varalle, jotka nyt epäonnistuvat SOAP-rajapinnan kautta tehdyissä kutsuissa.

Loppuviikosta keskityin HR-projektin Microsoft Forms muutoksiin. Tässä suurimman osan ajasta vei Power Automaten kehittämisessä jälleen vastaan tulleet virheet joko jo ennen ajon käynnistymistä tai ajon aikana. Kuten projektissa mukana ollut ulkopuolinen Power Apps konsulttikin sanoi, virheet ja niiden debuggaaminen on yksi osa Power Automate työnkulkujen kehityksessä. Olen huomannut aiemmin ja huomasi myös tätä ongelmaa selvittäessä, että jossain kohtaa, kun törmään jatkuvasti eri virheisiin ja varsinkin, jos törmään jatkuvasti samaan virheeseen, niin turhautumisen taso alkaa nousta. Silloin voisi olla hyvä hengähtää ja pitää tauko ongelman selvityksestä. Stressaantuneena ja väsyneenä ongelmanratkaisun yrittäminen saattaa jopa pahentaa tilannetta. (Cocca 2022.)

Tässä tapauksessa en pitänyt taukoa, vaan jatkoin googlaamista ja virheen pohtimista turhautumisesta huolimatta. Löysin lopulta ratkaisun ongelmaan ja sain Microsoft Formsin kautta lähetetyn ilmoituksen tiedot määpöytymään Power Automaten kautta oikein Power Appsiin. Vaikka tässä tapauksessa en perääntynytäkään vaan jatkoin eteenpäin, niin olen varmasti välillä myös perääntynyt ja tehnyt sen mahdollisesti myös tiedostomatta. Olen esimerkiksi saattanut vaihtaa hankalan ongelman selvityksestä toiseen tehtävään ja jatkaa selvitystä myöhemmin samana tai seuraavana päivänä. Voisinkin jatkossa pyrkiä paremmin tiedostomaan turhautumisen ja tehdä tietoisien valinnan, että jätän asian hautumaan ja palaan siihen seuraavana päivänä tai myöhempänä ajankohdantana uudestaan. Näin voisin tietoisella valinnalla pyrkiä optimoimaan ajankäyttöäni.

Kehittäjän roolissa vastaan tulleiden ongelmien selvityksessä olen luottanut vahvasti virheviestin tulkitsemiseen sekä intuitioon mahdollisista ratkaisuvaihtoehdoista. Virheviestin tulkitsemisen tulisi olla lähtökohta, mutta löytyy myös metodologisempia tapoja jatkaa virheen selvitystä siitä eteenpäin. Madison Kanna (2021) puhuu omassa artikkelissaan George Polyan 1940-luvulla kehittämästä ongelmanratkaisumenetelmästä, joka on edelleen laajasti käytetty menetelmä esimerkiksi ohjelmointiongelmien selvittämiseen ja niiden selvittämisen opettamiseen. Menetelmässä on neljä vaihetta, joista ensimmäisessä tunnistetaan ja ymmärretään ongelma, seuraavassa luodaan suunnitelma, joka kolmannessa vaiheessa toteutetaan ja viimeisessä eli neljännessä vaiheessa ongelmanratkaisua tarkastellaan kriittisesti ja pyritään löytämään kehityskohtia sekä optimoimaan ratkaisua. Voisin jatkossa testata myös tämän nelivaiheisen ongelmanselvitysmenetelmän toimivuutta omassa työssäni. Tiedostaa paremmin ongelmanselvityksen eri vaiheet ja varoa hyppäämästä johdopäätökseen, kuten ihmisen mieli välillä ihan automaattisesti saattaa tehdä.

Seuraavalla viikolla aion ottaa POC-hankkeen organisoitumisen osalta seuraavan vaihteen käyttöön ja käynnistää Scrum-kehityksen kaltaiset päivittäispalaverit (Daily Scrum), jotta pysymme kaikki paremmin kartalla missä mennään, mikä on tehtävien tilanne ja mitä tarvitsee vielä tehdä, jotta pääsemme hankkeessa eteenpäin.

3.2 Seurantaviikko 2 (vko 36)

Viikon tavoite: Tämän viikon tavoitteena on ottaa hieman järjestäytyneempi ote tehtävien tilanteesta ja suorittamisesta Scrum-tyyppisillä päivittäispalavereilla. Teknisinä tavoitteina on saada HR Häirintäilmoituksen raportointi kuntoon. Lisäksi sovellus tarvitsisi ominaisuuden, joka säilytysajan umpeuduttua anonymisoisi ilmoituksen luomalla sen uudestaan samoilla tiedoilla häivyttäen henkilötiedot ja tarkat kuvaukset. Tämän muutoksen tavoite on mahdollistaa ilmoitusten historian seuranta pidemmällä aikavälillä. POC-hankkeen osalta olisi tärkeää pystyä edistämään työtilauksen käsittelyn tapauksia ja niihin liittyviä muita pienempiä kehityksiä.

Maanantai 5.9.2022

Tavoite: Jatkaa POC-hankkeen muiden integraatioiden kehittämistä.

Toteutuminen: Integraatiokehittäjällä oli tullut ongelmia uusien rivien luomisessa Dataversen tauluun, johon on ajatuksena tuoda asiakkaan ERP-järjestelmästä henkilö- ja työntekijätiedot. Tietoja tarvitaan muun muassa työaikakirjausten siirron mahdollistamiseen, sillä ERP-järjestelmään tietoa luotaessa työntekijän HR-henkilönumero täytyy olla tiedossa. Uuden rivin luomisessa tuli virheilmoitus: *“User with ID d87ed1e0-75c0-ec11-983f-000d3a2d449b does not have Create permissions for the kict_first_name attribute in the kict_person_employee entity”*

Tällä kertaa virheilmoitus oli erittäin selvä ja sain suoraan virheilmoituksesta käyttäjän ID:n, jonka avulla pystyin tarkistamaan käyttäjän oikeuksia Dataversen puolella. Kävi ilmi, että integraatiokäyttäjältä puuttui kirjoitusoikeudet Dataversen uuteen tauluun. Käyttöoikeuksien lisäämisen jälkeen rivien luominen onnistui.

Tiistai 6.9.2022

Tavoite: Työtilaus- ja työaikakirjausten toimivuuden varmistamiseen liittyvien kehitysten edistäminen.

Toteutuminen: Pidimme päivittäispalaverin (Daily Scrum) integraatiokehittäjän kanssa. Nyt vuorostaan ratkaisuarkkitehti oli poissa sairastumisen vuoksi. Kävimme palaverissa tilanteen läpi ja katsoimme Jirasta tehtäviä sekä priorisoimme mitä tulisi tehdä seuraavaksi. Listalla oli muun muassa työtilauksen tiläkäsittelyyn liittyvä logiikkamuutos, jossa osa SOAP-rajapinnan kautta tällä hetkellä tehtävistä tilamuutoksista tehtäisiin jatkossa SQL-suorituksella suoraan kantaan hyödyntämällä asiakkaan ERP-järjestelmän metodeja. Tämän lisäksi listalta löytyi työtilauksen tilojen oikea järjestys integraatiossa. Huomasin testatessani työtilauksen tiloja, että käyttöliittymän kautta liian nopeasti tilaa vaihtaessa myöhemmin suorittanut integraatiotapahtuman lähetys saattaa jäädä kokonaan pois. Tähän pohdimme tilojen priorisointia ratkaisuksi, joka tarkoittaa, ettei seuraavaa tilaa

voitaisi siirtää eteenpäin ennen kuin edellisen tilan käsittely on valmis. Tämä toki tarkoittaa, että pitää tarkasti määritellä mistä tilasta siirrytään mihinkin tilaan eli mikä on tilojen oikea järjestys. Tätä täytyy vielä pohtia lisää, sillä kummallakaan ei vielä ollut selvää ratkaisua priorisoinnin toteuttamiseksi.

Keskiviikko 7.9.2022

Tavoite: Edistää POCin vaiheen 2 asioita mahdollisimman paljon, sillä huomenna on POCin jatko- ja seurantalaveri, joka pidetään asiakkaan kanssa noin kahden viikon välein. Tämän päivän edistettävänä asioina muun muassa työntekijä- ja kohdeintegraatiot sekä mobiilin työaikakirjausten offline-toiminnallisuus. HR Häirintäilmoituksen poistoon ja ilmoituksen anonymisointiin liittyvä kehitys olisi myös työlistalla tälle päivälle.

Toteutuminen: Aloitimme päivän integraatiokehittäjän kanssa päivittäispalaverilla (Daily Scrum). Hänellä oli haasteita parin eri integraatiokehityksen kanssa, jotka hän halusi palaverissa käydä läpi. Haasteet läpikäytyämme integraatiokehittäjällä vaikutti olevan selvät sävelet edellä mainittujen korjaamisesta ja sitä kautta hankkeen eteenpäin viemisestä.

Tein HR Häirintäilmoitukseen toiminnallisuutta, joka lisää valmiiksi kuitatulle ilmoitukselle poistopäivämäärän. Tein lisäksi Power Automateen työnkulun, joka käsittelee sellaiset ilmoitukset, joiden poistopäivämäärä on menneisyydessä. Kyseinen tehtävä vaati jonkin verran ohjemateriaalien ja muiden lähteiden tutkimista, jotta onnistuin siinä. Yksi toteutuksen keskiössä oleva uusi asia oli laskennalliset sarakkeet (calculated columns). Olin aiemmin pohtinut miten ne toimivat ja miten niitä voisi hyödyntää mutten ollut vielä kertaakaan sellaista aiemmin kokeillut tai käyttänyt. Niiden avulla pystyy luomaan kentän, jolle voi luoda esimerkiksi arvon ehtoon perustuen. Sain tehtyä ehdon, jonka mukaan poistopäivämäärään tulee päivämäärä ilmoituksen tilan sekä arkistointipäivämäärän mukaan. Mikäli kyseessä on poisto eli tila on muu kuin valmis, poiston säilytysajaksi on määritetty seitsemän päivää. Valmiiksi käsitellyn ilmoituksen automaattinen poisto käynnistyy valmiuskuitauksen myötä ja poistopäivämäärä muodostuu arkistointipäivämäärästä kahden vuoden päähän. Totesin laskennallisten sarakkeiden käytön helpoksi ja erittäin hyödylliseksi ja tulen varmasti jatkossakin hyödyntämään niitä sopivissa kohdissa.

Suunnittelin vielä myöhemmin POC-hankkeen työaikakirjausten offline-toiminnallisuutta ja jouduin sen kanssa kyllä todella pohtimaan, miten saan sen tehtyä. Aiemmin työaikakirjaus tallennettiin suoraan kantaan mutta pelkästään näin toteutettuna se aiheutti ongelman, jos käyttäjä sulki sovelluksen. Silloin muuttujaan tallennettu tieto hävisi sovelluksen sulkemisen yhteydessä eikä sovelluksen uudelleen käynnistämisen jälkeen enää ollut mahdollista yhdistää päällä olevaa työaikakirjausta käyttäjän tekemän toimenpiteen kanssa. Ongelman ratkaisun avaimet piilivät kokoelmassa

(collection), jonka avulla sain tallennettua käyttäjän kirjauksen ensin kokoelmaan, tallennettua tiedot laitteelle offlinea varten sekä viimeisenä Patch-funktion avulla tallennettua kokoelman kentistä tiedot tietokantaan. Näin tietojen pitäisi pysyä aina ajan tasalla huolimatta siitä suljetaanko sovellus välissä vai ei.

Torstai 8.9.2022

Tavoite: Päivä alkaa uudella, hyväksi havaitulla päivittäispalaverilla. Tavoitteena on saada viimeistelyä työaikakirjausten offline-toiminnallisuus sen demoamista varten sekä tarkistettua mitä vielä vaiheen 2 osalta tarvitsee tehdä. Tänään käydään myös asiakkaan business partnerin ja sovel-lusarkkitehdin kanssa läpi POC-hankkeen tilanne.

Toteutuminen: Aamulla tarkistimme integraatiokehitysten tilanteen ja teimme korjauksia muuta-miin integraatioihin muun muassa korjataksemme turhat 0-hintaiset tuntirivit, joita integraatio lähetti asiakkaan ERP-järjestelmään jokaisen tilanmuutoksen yhteydessä. Tuntirivejä tarvitaan vain, kun päivitetään työn valmistumista tai aloituksen jälkeistä keskeytystä, joissa tunteja on ehtinyt kerty-mään työlle. Mikäli tunnit ovat tyhjät ja päivitetään esimerkiksi vasta työtilauksen hyväksyntää tai aloitusta, silloin tunnit ovat turhat ja ne voi jättää kokonaan siirtämättä.

Offline-toiminnallisuus vaati vielä kirjausten välisen ajan näyttämisen tunteina ja minuutteina. Löy-sin tähän hyvän videon: "Power Apps Date Time Math - Do calculations for payroll and ti-mesheets", jossa Microsoft MVP Shane Young selitti funktion taustaan liittyvistä aikakäsittelyistä (Young 2021).

Jiran työlistalla oli myös tehtävä, jonka mukaan pitäisi saada push-viesti, kun uusi keikka on koh-distettu käyttäjälle. Lähdin tutkimaan push-viestimahdollisuutta YouTube-videoiden avulla ja löysin Microsoft MVP Reza Dorrnin tekemän videon: "Create push notifications for Power Apps mobile – Tutorial". Siinä selitettiin selkeästi, miten push-viesti lähetetään Power Automaten avulla käyttäjän puhelimeen. (Dorrani 2021a). Kokeilin soveltaa Rezan käyttämää menetelmää käyttäen siihen Da-taversen triggeriä ja sain push-viestin lähetettyä onnistuneesti omaan puhelimeeni.

Perjantai 9.9.2022

Ei opinnäytetyön kontekstiin liittyviä työtehtäviä

Viikkoanalyysi 2

Tällä viikolla osaamisen kehittämisen keskiössä oli ketterän kehityksen periaatteiden parempi omaksuminen sekä teknisen ymmärryksen kasvattaminen. Lanseerasin POC-hankkeemme tiimille päivittäispalaverit (Daily Scrum), joissa tavoitteena on käydä päivän kehitystavoitteet, mahdolliset

haasteet sekä eteneminen läpi. Palaverin jälkeen samana päivänä voidaan viestiä muun muassa ryhmäkeskustelussa sekä tarvittaessa ottaa Teams-puheluita. POC-hankkeessa käyttämäämme kehitysmenetelmää voisi kuvailla kevyeksi ketterän kehityksen malliksi. Siinä on jotain Scrumiin viittaavia piirteitä mutta kaikilta osin se ei ole organisoitunutta tekemistä, johtuen siitä, että olemme hankkeen kanssa vielä POC-vaiheessa. Scrumin perusmalliin kuuluu tuotteen kehitysajon (product backlog), josta valitaan tietyt tehtävät yhteen sprinttiin, jonka kesto tyypillisesti on kahdesta neljään viikkoa. Päivittäisissä palavereissa (Daily Scrum) priorisoimme tehtäviä ja jokaisen sprintin jälkeen pidämme sprintin retrospektiivin, jossa tarkastelemme sprintin onnistumista ja pyrimme löytämään kohdat, joissa on parannettavaa. (Schwaber & Sutherland 2020, 7–10.) Kehitysajon, jota priorisoimme päivittäin, löytyy Jirasta. Kehitysjaksot on jaettu vaiheisiin (phase 1, phase 2, phase 3...) ja jokaiselle vaiheelle on mietitty tavoitteet, jotka pyritään vaiheen aikana saavuttamaan. Vaiheen kestoa ei ole Scrumin mallin mukaisesti määritetty, vaan kesto riippuu tehtävien valmistumisesta.

Tarkistimme Power BI gurun kanssa HR Häirintäilmoitukseen liittyvän raportoinnin tilaa ja muutoksien toteutusta, joista olimme sopineet viime viikon perjantaina. Olin tehnyt tarkastelua varten demodataa, jonka avulla pystyimme varmistumaan rajauksien toimivuudesta. Kehitin tässä yhteydessä ymmärrystäni Power Apps:n ja Power BI:n linkittämisestä toisiinsa. Huomasimme muun muassa, että Power Appsia varten valmiiksi perustettuja Active Directory käyttöoikeusryhmiä (security group) voitiin hyödyntää Power BI-raportin käyttöoikeusrajauksessa. Power BI tunnistaa Power Apps:n kanssa samalla tavalla Office 365:n kautta käyttäjän ja osaa sitä kautta näyttää oikeat tiedot käyttäjälle.

Tällä viikolla oli useita mielenkiintoisia teknisiä haasteita Power Apps:n ja Power Automaten osalta kuten POC-hankkeen työaikakirjausten offline-toiminnallisuuden luominen, automaattisen push-viestin generoituminen ja HR-projektin ilmoitusten säilyvyyteen ja anonymisointiin liittyvä kehitys. Niiden avulla opin paljon uutta edellä mainittujen järjestelmien mahdollisuuksista sekä tavoista, joilla haasteita pystyy ratkomaan. Offline-toiminnallisuuden toteuttamisessa haastavinta oli ymmärtää, miten saan tallennettua kokoelmaan tallennetun kirjauksen tietokantaan samanlaisena ja latauksen yhteydessä linkitettyä kirjauksen tietokannan kirjaukseen, jotta sovelluksen sulkeminen välissä onnistui. Pääsin myös harjoittelemaan erilaisia aikamuunnoksia Power Appsisssa Shane Youngin videoiden kautta soveltamalla. Tarvitsin esimerkiksi työaikakirjausten välisen keston teollisuustunteina sekä tunteina ja minuutteina, jotta käyttöliittymässä tuntien näyttäminen olisi mahdollisimman selkeää käyttäjälle.

Automaattisen push-viestin logiikan luominen sujui melko kätevästi Reza Dorrnin videon avulla ja ymmärsin videon kautta logiikan, jolla Power Automate lähettää push-viestin puhelimeen. Ratkaisu oli hyvinkin helposti ymmärrettävissä (kuva 2). Ensiksi tarvitaan liipaisin triggeri, joka aktivoituu

aina kun sovelluksen tietokantaan tulee uusi rivi. Tämän lisäksi liipaisimen tunnistamasta rivistä haetaan käyttäjän sähköposti, joka tarvitaan "Send push notification V2" Power Apps toiminnon suorittamiseksi oikealle käyttäjälle. Valinnaisten parametrien avulla voidaan mahdollisesti avata juuri kyseinen työtilaus mobiilissa tai tehdä jokin muu toiminto sovelluksen avaamisen lisäksi.

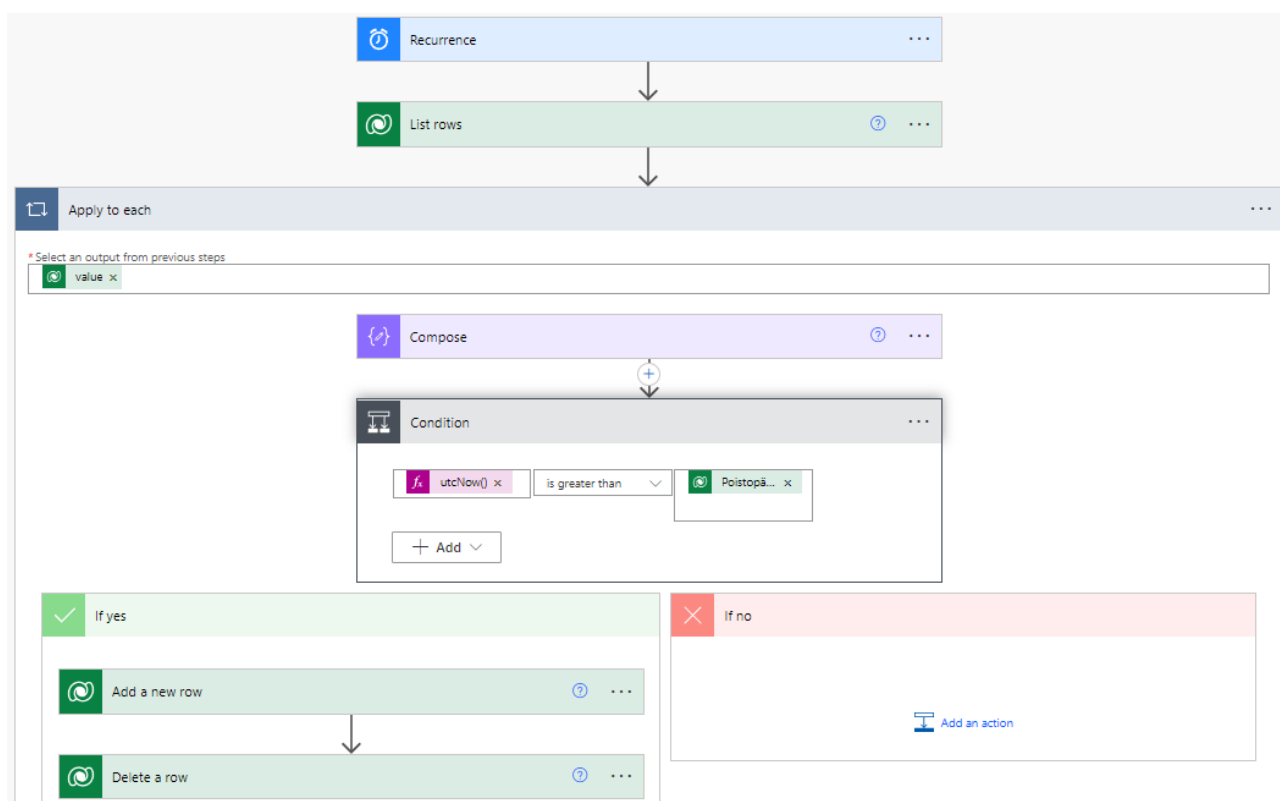
The image shows two steps in a Microsoft Power Automate flow:

- Step 1: When a row is added, modified or deleted**
 - * Change type: Added
 - * Table name: Work orders
 - * Scope: Organization
 - Show advanced options: [dropdown arrow]
- Step 2: Send push notification V2**
 - * Mobile app: Power Apps
 - * Your app: [redacted]
 - * Recipients Item - 1: User Email (with a trash icon)
 - + Add new item
 - * Message: "You have a new work order available"
 - * Open app: Yes
 - Parameters: Key value parameters to pass with the notification.

Kuva 2. Push-viesti uudesta työtilauksesta käyttäjän puhelimeen (Microsoft Power Automate 2022a)

HR-projektin säilytykseen ja anonymisointiin liittyvä Power Automate työnkulku oli hieman monimutkaisempi toteuttaa mutta loin sitä palasissa ja testasin jatkuvasti, jolloin pystyin reagoimaan tilanteisiin nopeasti. Kuvassa 3 näkyy työnkulun logiikka. Valitsin ensiksi ajoitetun triggerin, jonka

jälkeen hain Dataversen rivit ja niihin liittyvät tiedot. Kun tiedot olivat työnkulun tiedossa, kävin kaikki rivit läpi ja vertasin niiden poistopäivämäärää nykyhetkeen yksinkertaisella ehdolla. Mikäli nykyhetki on suurempi kuin poistopäivämäärä, ilmoitus muutetaan anonymiksi luomalla se vain historiadataan vaatimilla tiedoilla uuteen historiatauluun. Tämän jälkeen rivi poistetaan alkuperäisestä taulusta. Mikäli poistopäivämäärä on suurempi kuin nykyhetki, ajo vain terminoituu.



Kuva 3. Power Automaten työnkulku ilmoituksen säilytykseen liittyen (Microsoft Power Automate 2022b)

Huomasin kuvankaappauksia ottaessani, etten ollut nimennyt Power Automaten työnkulkujen eri vaiheita millään tavalla, vaan ne olivat vielä oletusnimillä. Olen keskittynyt nimeämiseen alusta asti Power Apps sovelluksen kehittämisen osalta ja pyrkinyt noudattamaan standardin mukaista nimeämistä kohdissa, joissa se on järkevää. Standardi nimeäminen helpottaa muun muassa sovelluksen luettavuutta, hallinnointia sekä jonkun toisen mahdollisuutta tukea tai kehittää sovellusta (Dunn 2018).

Voisin jatkossa kiinnittää parempaa huomiota Power Automaten työnkulkujen nimeämiseen, jotta niiden luettavuus ja ymmärrettävyys säilyy parempana ja tulevaisuudessa myös jonkun muun on helppo niitä ymmärtää ja tarvittaessa muokata.

3.3 Seurantaviikko 3 (vko 37)

Maanantai 12.9.2022

Tavoite: POC-hankkeen toisen vaiheen keskeneräisten asioiden edistäminen. Lisäksi ajatuksena pohtia hieman sovelluksen versionhallintaa. Tällä hetkellä sovellus elää Power Appsin sisällä sekä satunnaisesti otetuissa ZIP-tiedostovienneissä asiakkaan OneDrivessä sekä omalla koneellani.

Toteutuminen: Tein parannuksia Power Appsin työaikakirjaustoiminnallisuuteen, jotta voisimme jatkossa lähettää niitä Azureen ja sitä kautta asiakkaan ERP-järjestelmään. Tutustuin myös Git-versionhallintaan Timo Pertilän (2022) artikkelin avulla ja päätin kokeilla GitHubin yhdistämistä Power Appsiin. Tämä ominaisuus oli julkaistu käyttäjille tämän vuoden alussa. Ensimmäiset tuntevat GitHubin yhdistämisestä Power Appsiin olivat positiiviset. Yhdistäminen oli todella helppoa ja muutokset tallentuivat kätevästi GitHubiin.

Tiistai 13.9.2022

Tavoite: Tavoitteena tutkia miten havaitsemani Power Apps mobiilisovelluksen hitauteen voi vaikuttaa. Hyvä lähtökohta on tutustua tarkemmin Microsoftin omaan dokumentaatioon suorituskyvyn parantamisesta. HR Häirintäilmoituksen osalta tavoitteena on tuotantovalmiuden lähestyessä selvittää mitä kaikkea tuotantoon siirtyminen tarkoittaa ja mitä kaikkea sitä varten täytyy tuotantoympäristöön perustaa. POC-hankkeen osalta pyrimme jatkamaan työaikakirjausten ja työtilausten käsittelyn sujuvoittamista.

Toteutuminen: Kävimme aamulla päivittäispalaverissa läpi päivän asioita. Integraatiokehittäjä työskentelee myös tuotantohaasteiden kanssa ja asiakkaalla on ollut nyt muutaman viikon ajan hankaluuksia yhden tuotantointegraation kanssa, joka vaikuttaa kehitykseemme siten, että integraatiokehittäjän täytyy välillä osallistua ongelmanselvityksiin tuotannossa, joka on aina POC-hankkeemme kehityksestä pois. Tämän vuoksi on tärkeää myös edistää muiden jäsenten oikeuksia kehitysympäristöön. Kyselin saman tien asiasta asiakkaan ICT:n palvelun toimituksesta vastaavalta henkilöltä.

Aloitimme tällä viikolla kehitystyön työaikakirjausten lähettämisestä asiakkaan ERP-järjestelmään Azure Logic Appsin kautta. Kävimme tänään integraatiokehittäjän kanssa Azuren puolen määppäyksen läpi ja huomasimme, että muutama työaikakirjausten lähettämiseen tarvittava kenttä puuttui Dataversen tietokannasta. Tein päivitykset tietokantaan, jonka jälkeen pystyimme määppäämään Azuren puolella kaikki muut kentät oikeisiin arvoihin paitsi rowid, rowkey ja rowversion. Edellä mainitut kentät ovat asiakkaan ERP-järjestelmän Oracle tietokannan avainkenttiä, jotka ovat pakollisia

lisäyksen ja muutoksen yhteydessä. Selvittelimme asiaa integraatiokehittäjän ja tietokantaspecialistin kanssa muttei vielä tänään löydetty sopivaa ratkaisua ongelmaan.

Varasin myös ulkopuoliselta Power Apps konsultilta seuraavaksi päiväksi aikaa käydä läpi HR Häirintäilmoituksen tuotantosiirtoon liittyviä asioita.

Keskiviikko 14.9.2022

Tavoite: Käydä läpi tuotantoon siirtoon liittyvät asiat ulkopuolisen Power Apps konsultin kanssa ja sen jälkeen varmistaa ICT palvelutoimituksesta vastaavalta henkilöltä, että kaikki tarvittavat asiat saadaan edistettyä ennen 1.10. olevaa tuotantovalmiuden päivämäärää. POC-hankkeessa tavoitteena jatkaa työtilauksen tilojen selvittämistä.

Toteutuminen: Aloitimme aamun päivittäispalaverilla, jossa keskityimme heti eilen kesken jääneen työaikakirjauksen Oracle-kyselyn selvittämiseen. Olimme eilen todella lähellä ratkaisua mutta emme saaneet työaikakirjausta siirtymään siitäkään huolimatta, että ERP-järjestelmä palautti vastauksena "200 – OK", eli lähetys vaikutti menevän perille asti. Tutkimme yhdessä integraatiokehittäjän ja tietokantaspecialistin kanssa tilannetta. Ongelmana vaikutti olevan automaattisesti generoituvat kentät, jotka ERP-järjestelmä luo automaattisesti uuden rivin luomisen yhteydessä, mutta jotka Azuressa olivat pakollisia eli vaativat jonkun arvon, jotta Azuren työnkulun pystyi tallentamaan. Testailimme erilaisia variaatioita kuten eri muotoja nykyhetken aikaleimasta rowversion kenttään, satunnaista numeroa rowid-kenttään, GUID-funktion luomaa globaalia tunnistetta, tyhjää tekstiä sekä kenttien poistamista lähtevästä viestistä kokonaan. Mikään edellä mainituista ei aluksi tuntunut auttavan mutta lopulta löysimme ratkaisun, jossa rowversion muutettiin vastaamaan ERP-järjestelmän sysdate -kenttää, rowid lähetettiin tyhjänä ja answerid haettiin ERP-järjestelmästä SQL-lauseen avulla.

Päivittäispalaverin jälkeen kävin keskustelua ulkopuolisen Power Apps konsultin kanssa HR Häirintäilmoituksen tuotantoon siirtämisestä. Läpikäynnistä sain varmistuksen oman ymmärryksen tueksi sekä muutaman hyvän vinkin kuten Power Automate työnkulkujen tekemisen suoraan ratkaisuun, jolloin kaikki tarvittavat komponentit, yhteydet ja viittaukset ovat kunnossa ja siirtyvät ratkaisun mukana ympäristöjen välillä. Tämän lisäksi hän ehdotti, että kehitysympäristöön kannattaisi luoda uusi AD-käyttöoikeusryhmä, joka estäisi tuotantokäyttäjiä vahingossa käyttämästä kehitysympäristöä tuotannon sijaan. Yritin palaverin jälkeen olla yhteyksissä asiakkaan ICT:n palvelun toimituksesta vastaavaan henkilöön mutten saanut häntä kiinni. Asialla on kiire sen vuoksi, että hän on jäämässä ensi viikolla lomalle, joten asia täytyisi saada hoidettua mahdollisimman pian, jotta luvattu tuotantovalmius on mahdollinen heti lokakuun alusta.

Aiemman menestyksen siivittämänä jatkoimme muiden POC-hankkeen haasteiden ratkomista ilta-päivällä ja keskityimme seuraavaksi epäonnistuneiden tilojen muutoksiin. Loppupäivä meni suurimaksi osaksi tilanmuutosten selvittämiseen sekä muutamiin käyttöliittymän parannuksiin. Päivän päätteeksi meillä oli tilanne, jossa ainoastaan ”pidossa” -tilanmuutos oli ratkaisematta. Päivä oli kokonaisuudessaan erittäin tuottoisa ja opin paljon uutta Oraclen tietokantasuorituksista Azuren kautta. Puhuimme että olisi hyvä ottaa myös ”sisäisesti” kehitystiimin sekä asiakkaan kumppaniyrityksen BAS-tiimin kanssa demo ennen asiakkaalle esittämistä. Tällä pyrkimys olisi varmistaa sitä, että demo asiakkaalle on mahdollisimman sujuva.

Torstai 15.9.2022

Tavoite: Tavoitteena on selvittää viimeisen tilanmuutoksen tekeminen Azuressa, jonka jälkeen sovellus alkaa olla hyvässä vaiheessa asiakkaan liiketoimintajohdon esitystä varten. Tämän lisäksi pyrin edistämään HR Häirintäilmoituksen tuotantoon siirtoon liittyvien asioiden hoitamista asiakkaan ICT:n palvelun toimituksesta vastaavan henkilön kanssa. Jatkan tänään myös Power Apps sovelluksen hitaan avautumisen selvittämistä. Huomasin jo, että tietojen rajaaminen auttoi jonkin verran. Concurrent-funktio on yksi varteenotettava, jota pitäisi tutkia lisää. Sen avulla tietyt toimenpiteet saadaan suorittamaan samanaikaisesti sen sijaan, että ne suorittaisivat yksitellen peräjälkeen.

Toteutuminen: Tämän päivän osalta toteuma ei kyllä aivan vastannut tavoitteita. Aamu meni asiakkaan tuotannossa ilmenneen akuutin virhetilanteen selvityksessä, joten emme pystyneet pitämään päivittäispalaveria normaalisti. Emme päässeet tutkimaan viimeistä tilanmuutoksen toteuttamista koko päivänä, sillä integraatiokehittäjän, tietokantaspecialistin ja minun kalenterini menivät täysin ristiin. Sovimme, että jatkamme selvitystä ensi viikolla, sillä olen lomalla huomisen päivän.

Meillä oli sovittuna HRIS-specialistin ja HR työsuhde- ja työsuojelupäällikön kanssa läpikäynti, jossa tarkistimme HR Häirintäilmoituksen viimeisimmän tilanteen. Näytin muun muassa arkistointitoiminnallisuden, puhuimme tuotantoon siirrosta, sovelluksen käyttökoulutuksen pitämisestä sekä muutamista pienistä muutoksista, joita huomasimme läpikäynnin yhteydessä. HRIS-specialistin kanssa jäimme vielä palaverin jälkeen hetkeksi linjoille ja tarkistimme hänen pitämän Excel-taulukon avulla, että kaikki tarvittavat tehtävät ovat joko hoidossa tai jo hoidettu.

Tänään oli tavoitteena saada tuotantoon siirtoa edistettyä muun muassa selvittämällä miten saisin tuotantoympäristön luotua. En kuitenkaan pystynyt sitä edistämään, koska ICT:n palvelun toimituksesta vastaava henkilö oli poissa tänään. Kysyin hänen kollegaltansa, että onko hänellä tietoa asiakkaan tavasta järjestää Power Apps ympäristöjä sekä AD-ryhmiä. Tarvitsisin kehitysympäristön lisäksi tuotantoympäristön sekä muutaman AD-ryhmän, jonka avulla pystyn rajaamaan pääsyjä

sekä tuotannossa että kehitysympäristössä. Tällä ei kuitenkaan enää ole niin palava kiire, sillä sovimme HR:n kanssa pitämässämme palaverissa, että sovellus olisi tuotantovalmis lokakuun puolessa välissä, jonka jälkeen testaisimme sovellusta tuotannossa ja 27.10. olisi yhteinen koulutus sovelluksen käyttäjille ja muille asiaan liittyville. Sovelluksen go-live eli käyttöönotto olisi heti marraskuun ensimmäisenä päivänä.

Ehdin jonkin verran tutkimaan myös Power Apps sovelluksen hitauteen ja suorituskykyyn liittyviä asioita ja löysin useamman hyvän videon, jotka Microsoftin oman dokumentaation ohella antavat paljon eväitä sovelluksen suorituskyvyn parantamiseen. Esimerkiksi Concurrent-funktio on yksi avaintekijöistä, kun puhutaan sovelluksen suorituskyvystä (Microsoft 2022c). Concurrent-funktion avulla tietyt suoritukset kuten muuttujien määrittely sekä kokoelmien luominen saadaan suoritettua samanaikaisesti. Samanaikaisesti suorittaminen säästää huomattavan määrän aikaa verrattuna siihen, että suoritukset menisivät peräjälkeen. (Dorrani 2021c.) Tämän lisäksi muun muassa tietokantakutsujen määrä ja laatu vaikuttavat suorituskykyyn. Yhtenä esimerkkinä ”viivytä tulostetta” -asetus, joka viivästyttää tietokantahaun suorittamista siihen asti, kunnes käyttäjä lopettaa kirjoittamisen. Ilman kyseistä asetusta, jokainen kirjaimen kirjoitus suorittaa tietokantahaun ja näin ollen hidastaa sovelluksen suorituskykyä huomattavasti. (Dorrani 2021b.) Tätä täytyy vielä tutkia enemmän seuraavalla viikolla ennen liiketoimintademoa, jotta voin demossa kertoa, että mitä asioita voidaan vielä parantaa, mikäli POC-vaiheesta jatketaan eteenpäin. Kaikkia parannuksia ei ole järkevää tehdä vielä, mutta Concurrent-funktion aion ottaa heti käyttöön.

Perjantai 16.9.2022

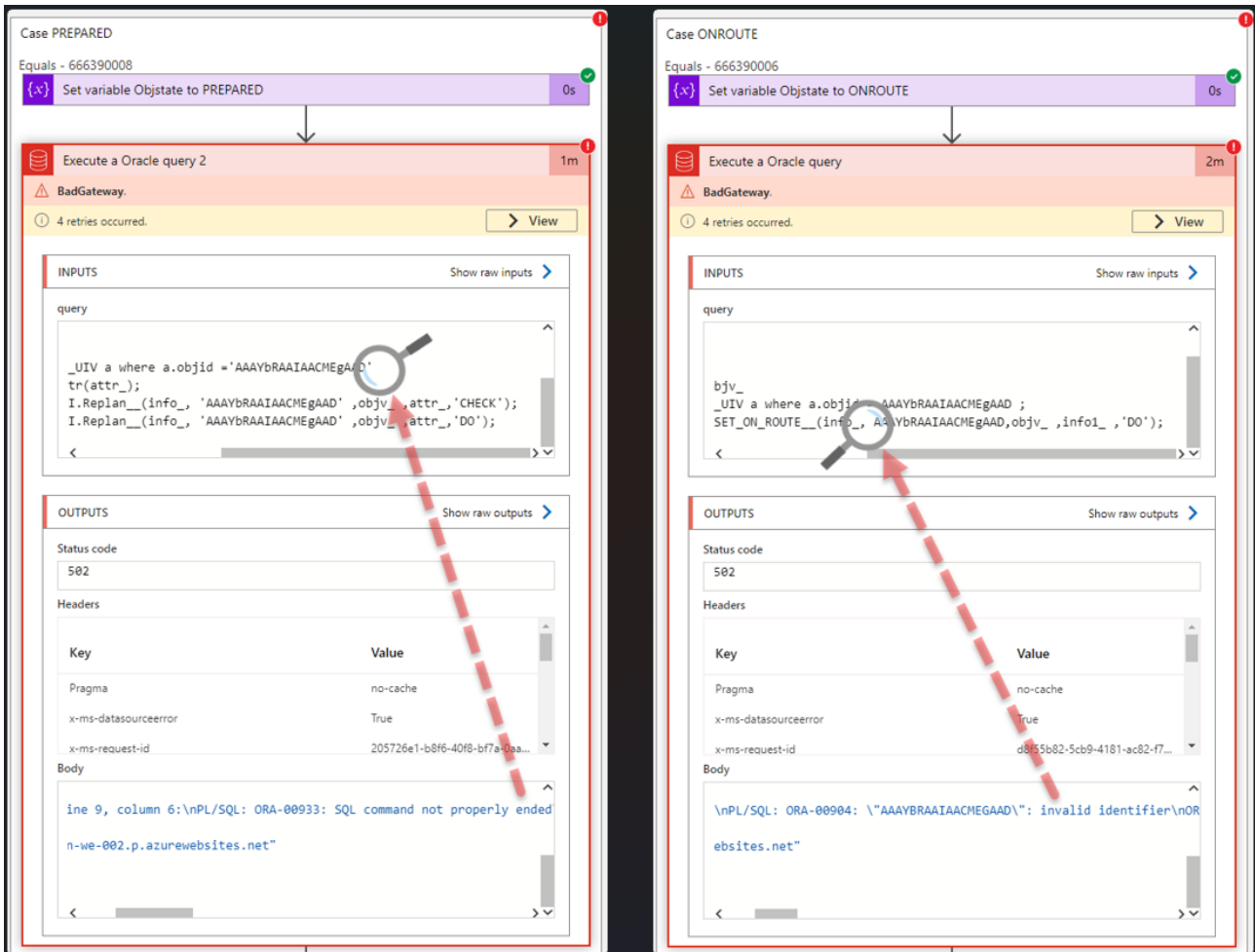
Vapaapäivä

Viikkoanalyysi 3

Mennyt viikko oli kiireinen ja etenkin POC-hankkeen tekniset haasteet veivät tämän viikon työajasta leijonan osan. Tämän viikon opit osuivat siis vahvasti teknisen kehittymisen tavoitteisiin. Alkuvuikosta jatkoimme työtilauksen tilanmuutosten selvittelyä ja iso osa ajasta kului Azuren integraation kehittämisessä ja debuggaamisessa. Etenkin Oraclen tietokantakyselyt tuottivat hankaluuksia jopa tietokantaspecialistille. Oli hyvä, että meitä oli useampi selvityksessä mukana, sillä jokainen ymmärsi jonkin verran selvitykseen liittyvistä eri osa-alueista ja yhdessä saimme muodostettua hyvän yhteisymmärryksen ja lopulta saimme ongelmat ratkottua.

Oraclen kanssa sai olla tarkkana ja opimme sen kantapään kautta, vaikka tiesinkin SQL:n syntaksin olevan herkkä virheille. Uskon myös tietokantaspecialistin olevan erittäin selvillä SQL-syntaksista kaikkine pilkku-, puolipiste- ja välimerkkisääntöineen. Kuvan 4 vasemmanpuoleisessa osiossa

suurennuslasi näyttää ”SQL command not properly ended” -virheen kohdan, joka aiheuttaa kyseisen virheen. Rivin lopusta puuttuu puolipiste. Kuvan 4 oikeanpuoleinen kuva näyttää vielä viheliäisemmän ja hankalammin löydettävän virheen, jossa suurennuslasin osoittamassa kohdassa on ylimääräinen väli heti pilkun jälkeen. Sen vuoksi ajo antoi virheeksi ”invalid identifier”, koska ajo ei tunnistanut objid:tä, jonka alussa on välilyönti.



Kuva 4. Oracle tietokantakyselyn virheet (Microsoft Azure Logic Apps 2022a)

Lähdimme mielestäni aluksi liian kunnianhimoisesti liikkeelle ja yritimme korjata liian monta asiaa virheestä kerrallaan, joka johti joko saman virheen toistumiseen tai uusiin virheisiin. Jonkin aikaa virhettä tutkittuamme ehdotin, että voisimmeko pilkkoa virheen tutkimista pienempiin osiin. Kuten jo aiemmassa viikkoanalyysissä mainitsin, on tehokkaampaa ja helpompaa debugata pieniä osia kerrallaan. Korjaamalla yhden asian kerrallaan ja testaamalla heti korjauksen vaikutusta suoritukseen, on huomattavasti helpompi ymmärtää mitä tapahtuu, mikä mahdollisesti menee pieleen ja miksi. Löysin artikkelin (Geeksforgeeks 2022), jossa tuodaan esille 10 erittäin tärkeää pointtia koodin debuggauksesta. Usein kehittäessä on mahdollista tulostaa suoritettavan koodin tuottama tulos näh-

täväksi, joka mahdollistaa kehittäjälle paremman ymmärryksen koodin tuottamasta lopputuloksesta. Sama asia pätee myös Logic Appsin työnkulkuihin, joissa Compose-toiminnolla voidaan näyttää ja tarkistaa tiettyjä osia työnkulusta. Tätä olen myös hyödyntänyt esimerkiksi Power Automate työnkulkujen kanssa ja olen huomannut integraatiokehittäjän kanssa pidetyissä palavereissa myös hänen käyttävän Compose-toimintoa hyödyksi koodin tulkitsemisessä.

Samassa artikkelissa puhuttiin myös automaattisista testeistä. Selvitin heti artikkelin luettuani, että onko testejä mahdollista tehdä myös Power Appsin puolella ja löysin senioritason ohjelmapääällikön kirjoittaman artikkelin Power Appsin testausstudiosta. Aion jollakin viikolla paneutua tarkemmin automaattisten testien luomiseen. Artikkelin lukemalla ymmärsin kuitenkin, että testien tekeminen on yritetty tehdä Power Appsissa mahdollisimman helpoksi. Sovelluksen toimintoja voi testausstudiossa nauhoittaa testitapauksiksi suorittamalla toiminnot nauhoitteen aikana sovelluksessa. Nauhoitettuja testitapauksia voi myös helposti muokata ja testitapausten suorittamisella voidaan todeta sovelluksen toimivuus sekä löytää mahdollisia bugeja, jotka voidaan korjata jo ennen, kun käyttäjä niitä huomaa. (Heaney 2020.)

Onnistuimme vahingossa aiheuttamaan ongelmia Logic Appsin puolelle vaihtamalla yhden työnkulun triggeriä esiversio triggerillä, joka ei lopulta toiminutkaan odotetusti. Emme huomanneet asiaa heti vaan ehdimme debugata toista virhettä hyvän tovin ennen tämän virheen huomaamista. Tässä on yksi esimerkki, jossa automaattinen testaus voisi auttaa. Henry Liun julkaisemassa yhteisön kirjoituksessa esitetään tapa, jolla Logic Appsia voidaan automaattisesti testata. Aiemmin automaattisen testauksen luominen on ollut hankalaa mutta vuonna 2021 Microsoftin julkaiseman ominaisuuden avulla testausta voidaan tehdä myös Logic Appsissa vaivattomammin. (Liu 2021.)

Tällä viikolla testailin myös Git-versionhallinnan yhdistämistä Power Apps sovellukseen. Maanantaina otin POC-hankkeen Power Appsin asetuksista Git-versionhallinnan yhdistämisen käyttöön ja yhdistin sovelluksen GitHubiin. Aluksi kaikki vaikutti toimivan hyvin, mutta huomasin viikon edetessä, etten olekaan aivan niin tyytyväinen Git-versionhallinnan toimivuuteen, kun maanantaina ensimmäisten havaintojen perusteella oletin. Power Apps muun muassa kysyi aina sovelluksen käynnistyksen yhteydessä Gitiä varten luotua suojausavainta, joka tuntui turhalta ja ylimääräiseltä vaiheelta. Tämän lisäksi huomasin pariin kertaan, että osa kehittämistäni asioista mystisesti katosi, vaikka olin tallentanut sovelluksen normaaliin tapaan. Ilmeisesti GitHubin ja Power Appsin välinen synkronointi olisi pitänyt muistaa tehdä joka kerta tallentaessa. Keskustelin myös ulkopuolisen Power Apps konsultin kanssa ja hän oli samaa mieltä, ettei Git-versionhallinta vielä ole sillä tasolla, että sitä voisi ainakaan tuotannossa käyttää. Samaa mieltä näytti olevan myös Timo Pertilä (2022),

jonka artikkelista: "Power Apps and Git version control" löysin hyvät ohjeet Git-versionhallinnan ko-keilemiseen sekä yhteenvedon hänen arviostaan Git-versionhallinnan käytöstä Power Appsin kanssa.

3.4 Seurantaviikko 4 (vko 38)

Maanantai 19.9.2022

Tavoite: Tälle päivälle on varattu ICT:n tilannekatsaus palaverin jälkeen aikaa ratkaisuarkkitehdin kanssa POC-hankkeen liiketoimintademoa varten. Tavoitteena saada liiketoimintademon esitysmateriaali tehtyä. Lisäksi tavoitteena olisi jatkaa POC-hankkeen työtilauksen käsittelyn viimeistä tapusta, jota ei viime viikolla pystytty jatkamaan risteävien aikataulujen takia. HR Häirintäilmoituksen osalta tavoitteena on läpikäydä tuotantoon vientiä varten tarvittavat asiat ICT:n palvelun toimituksesta vastaavan henkilön kanssa.

Toteutuminen: Päivittäispalaveri integraatiokehittäjän kanssa jäi tänäänkin välistä, sillä hän oli kiireinen tuotannossa ilmenneiden haasteiden vuoksi. Kävimme ratkaisuarkkitehdin kanssa tilanteen lyhyesti läpi ja totesimme, että olemme hyvässä tilanteessa liiketoimintademoa ajatellen. Tilanteen tarkistuksen jälkeen siirryimme pohtimaan liiketoimintademon esitysmateriaalin sisältöä. Jo aiemmin uusi business partner oli ohjeistanut meitä tuomaan esille, että sovellus on vasta POC-vaiheessa ja nyt esitettävä käyttöliittymä on vain hahmotelmamme siitä. Hänen mielestään on etenkin tärkeää saada liiketoiminnat ymmärtämään, että heillä on valta ja mahdollisuus suunnitella sovellus juuri kuin he sen haluavat. Esityksessä pyrimme tuomaan esille POCin käynnistyksen taustan, teknisen toteutuksen laajuuden, ratkaisun hyödyt sekä lisensseihin liittyviä arvioita. Tunnistimme esitystä tehdessä, että meidän olisi syytä olla myös tietoisia lisenssien hinnoista ja pystyä antamaan jonkinlainen karkea arvio kustannuksista.

Sain vihdoin myös sovittua asiakkaan ICT:n palvelun toimituksesta vastaavan henkilön kanssa palaverin, jossa kävimme heti tänään yhdessä läpi tuotantoon vientiin liittyviä asioita. Hän pystyi saman tien luomaan palvelutilin (service accountin), jolle kaikkien HR Häirintäilmoituksen komponenttien omistukset on tarkoitus keskittää. Saimme palaverin aikana myös luotua uuden tuotantoympäristön, joten suurin osa asioista selvisi yhden nopean palaverin aikana. Ainoastaan palvelutilin lisensseistä jäi selvitetäväksi ja pohdittavaksi, että tarvitaanko palvelutilille myös Power BI PRO-lisenssi.

Minulla on hyvät yhteydet aiempaan toimialan business partneriin, jonka avulla sain tietoja nykyisen ERP-järjestelmän lisenssien hinnoista ja hinnoittelusta. Keskustelun pohjalta päätelmänä on,

että sovelluksen kustannusten ei pitäisi ainakaan ylittää nykyisen ERP-järjestelmän mobiilisovelluksen kustannuksia. Parhaassa tapauksessa asiakas saa Microsoftin kanssa neuvoteltua hyvän diilin ja saa lisenssit vielä edullisempaan hintaan.

Tiistai 20.9.2022

Tavoite: Eilisen lisenssiselvityksen läpikäynti ratkaisuarkkitehdin kanssa sekä viimeisen työtilauksen käsittelytapauksen selvittäminen. Sain myös ajatuksen, jossa DS toimisi Power Appsin kanssa suoraan ilman suurempia konfiguraatiotarpeita. Tätä olisi hyvä pohtia lisää, sillä DS on erittäin tärkeä komponentti asiakkaan liiketoimintakriittisen prosessin osalta.

Toteutuminen: Kävin lisenssiasiaa ratkaisuarkkitehdin kanssa läpi ja tulimme samaan tulokseen kuin aiemman toimialan business partnerin kanssa eli sovelluksen kustannukset pitäisivät olla likimain samat mitä nykyisen mobiilisovelluksen kanssa ne ovat. Asiakkaalla on paljon muitakin Microsoftin tuotteita jo käytössä, joten voisi kuvitella, että Power Appsin sovelluskohtaisen lisenssin (per app plan) listahintaa varmaan saisi yli tuhannen käyttäjän tilauksella huomattavasti laskettua. Päätimme siis markkinoida/myydä sovellusta modernina sovelluksena, jota on nopea ja kevyt kehittää ja jonka pitäisi olla lisenssikustannusten osalta suurin piirtein saman hintainen nykyisen ratkaisun kanssa.

Keskiviikko 21.9.2022

Tavoite: Käydä POC-hankkeen asiakkaan kanssa käytävässä statuspalaverissa läpi hankkeen tilanne sekä esittää liiketoimintajohdolle valmisteltu Power Point -materiaali. Eilisen ajatuksen juoksun pohjalta tavoitteena myös testata voisiko DS (Dynamic Scheduling) eli asiakkaan automaattinen töiden optimointijärjestelmä toimia Power Appsin kanssa yhteen ilman suurempia konfiguraatiotarpeita. Tämä ei varsinaisesti vielä toiseen vaiheen scopeen kuulu, mutta on nopea testata ja erittäin hyödyllinen tieto liiketoimintademoa silmällä pitäen.

Toteutuminen: Kävimme päivittäispalaverissa ratkaisuarkkitehdin, integraatiokehittäjän sekä parin uuden henkilön kanssa läpi työtilausten oikean järjestyksen toteuttamisen vaihtoehtoja. Toinen ja omasta mielestäni parempi vaihtoehto on toteuttaa järjestyksestä huolehtiminen Azuressa Microsoftin Service Busia hyödyntäen. Service Busin hyödyntämisellä haetaan sitä, että kaikki yhteen työtilaukseen kuuluvat tapahtumat saataisiin menemään järjestyksessä, mutta tästä huolimatta muut työtilaukset voisivat suorittaa samanaikaisesti pitäen integraation suoritusnopeuden hyvänä. Toinen vaihtoehto olisi tehdä viivästyslogiikka Power Appsin puolelle. Huonoin vaihtoehto, jolla tilanne saataisiin varmasti ratkaistua, olisi muuttaa integraatio suorittamaan kutsut peräkkäin samanaikaisen suorittamisen sijaan. Tämä melko varmasti hidastaisi integraatiota liikaa, joten pyrimme ainakin ensiksi korjaamaan tilanteen jommallakummalla aiemmin esitetyistä vaihtoehdoista.

Kävin eilen mielessä läpi, että kuinka Power Apps voisi toimia lähes suoraan DS:n kanssa yhteen ja olin erittäin innokkaana tänään kokeilemassa, että toimisiko ajatus käytännössä. Meillä on alusta asti ollut olettamuksena, että DS:n ja Power Appsin yhdistäminen voi olla haastava juttu, jota tarvitsee käydä läpi myös sovellustoimittajan kanssa eikä siihen ollut vielä suuremmin ehditty keskittymään. Ajatuksena oli ollut tutkia DS:ää seuraavan vaiheen yhteydessä. Se on kuitenkin tärkeä osa kokonaisuutta, sillä asiakas pyrkii siihen, että mahdollisimman paljon töitä ohjattaisiin DS:n kautta manuaalisen ohjaamisen sijasta. Näin nimenomaan toimitaan nykyisenkin sovelluksen kanssa.

Torstai 22.9.2022

Tavoite: Tänään on tavoitteena saada uuden työtilauksen siirtyminen toimimaan asiakkaan ERP-järjestelmän suuntaan. Aion myös tänään tutkia sovelluksen hitaan avautumisen ongelmaa Polyan nelivaiheisen ongelmanselvitysmetodin avulla (Kanna 2021). Pidän myös HRIS-spesialistille lyhyen koulutuksen Power Appsin perusteista, jotta hän voi tarvittaessa jatkossa muokata muun muassa käyttöliittymän visuaalista ilmettä ilman minuakin.

Toteutuminen: Kävimme läpi uuden työtilauksen siirtymisen tarpeen ja hankkeeseen mukaan tulut uusi integraatioasiantuntija otti homman hoitaakseen.

Pidin lyhyen ja improvisoidun koulutuksen Power Appsin muokkaamisesta. Kävin perusteita läpi ja näytin joitakin tapauksia, joissa muokkauksia tehdään. Olisin voinut varautua koulutukseen huomattavasti paremmin, tehdä materiaalia valmiiksi ja suunnitella paremmin mitä koulutuksessa käyn läpi. Siihen ei kuitenkaan valitettavasti työtehtävien priorisoimisen vuoksi ollut mahdollisuutta.

Perjantai 23.9.2022

Tavoite: Tavoitteena ymmärtää paremmin miten eri aspektit vaikuttavat POC-hankkeen mobiilisuorituksen suorituskyykyyn.

Toteutuminen: Päivittäispalaverissa kävimme ratkaisuarkkitehdin, integraatiokehittäjän sekä hankkeeseen mukaan liittyneiden uusien integraatioasiantuntijoiden kanssa POC-hankkeen tilannetta läpi. Asiakas oli pyytänyt meitä rajoittamaan *person_employee* tauluun tuotavia tietoja GDPR:n vuoksi. Alun perin testausta helpottamaan olimme tuoneet tauluun myös etunimen, sukunimen, sähköpostin ja puhelinnumeron. Asiakkaan pyynnöstä päädyimme ottamaan muut kentät pois integraatiosta sähköpostikenttää lukuun ottamatta. Sitä tarvitaan käyttäjän yhdistämiseen Power Appsin puolella ja se on muutenkin jo Office 365:n kautta User-funktion avulla käytettävissä Power Appsin puolella. Huomasimme samassa yhteydessä, että nykyisellään integraatio toi kaikki käyttäjät päivittäin uudestaan, jonka vuoksi tauluun oli muodostunut duplikaatteja ja rivien määrä

oli kivunnut yli 18 000:een, vaikka taulussa pitäisi olla vain reilu 1000 käyttäjää. Ratkaisimme ongelman lisäämällä integraation aikaleima kentän samalla tavalla, kun aiemmin ja sen avulla suodattamalla siirrettävät tiedot ainoastaan muuttuneisiin tai uusiin käyttäjiin.

Uusi integraatioasiantuntija oli ottanut eilen työn alle uuden työtilauksen luomiseen liittyvän integraation. Hän ilmoitti saaneensa integraation valmiiksi ja että sitä voisi testata sopivassa välissä. Integraatiossa oli yritetty käyttää aiemmin kokeilemaamme "add, modify, delete" -triggeriä, jota ei viimeksi saatu toimimaan samassa yhteydessä. Olin siis heti testauksen alussa melko skeptinen sen toimimisen mahdollisuudesta. Totesimme nopeasti, ettei se toimi tälläkään kertaa ja hetken asiaa pohdittuamme päädyimme siihen, että integraatioasiantuntija korjaa työnkulun ja kokeilemme sen jälkeen uudestaan. Mielestäni paras ratkaisu olisi ollut erottaa uusi ja muokkaus omiksi työnkuiluiksi, mutta jostain syystä integraatiokehittäjät halusivat vielä kokeilla upottaa molemmat yhteen työnkulkuun. Mahdollisesti ajatuksena oli vähentää ylläpidon tarvetta lisäämällä molemmat samaan työnkulkuun.

Loppupäivän käytin eilen aloitetun suorituskyvyn muokkaamiseen ja testaamiseen.

Viikkoanalyysi 4

Mennyt viikko oli asiarikas, mielenkiintoinen ja laajalti opettavainen. Oman osaamiseni kehittämisen kohokohtia olivat sovelluksen suorituskykyyn vaikuttavien asioiden ymmärtäminen, siihen liittyvä tarkka testaaminen ja dokumentoiminen Polyan mallia (Kanna 2021) soveltaen, asiakkaan GDPR:ään liittyvien vaatimusten huomioiminen integraatiossa, HR:n vaatimusten ymmärtäminen HR Häirintäilmoituksen käyttöoikeuksien osalta, automaattisen optimoinnin (DS) yhteensovittaminen POC-hankkeen mobiilisovelluksen kanssa sekä liiketoimintademoon liittyvien oleellisten asioiden kuten kustannusten ja hyötyjen kartoittaminen.

Lähestyvää liiketoimintademoa silmällä pitäen POC-hankkeen teknisen toteutuksen tila näytti hyvältä ja rupesimme suunnittelemaan esitysmateriaalia ensi torstaina pidettävään demoan. Demon lisäksi esitykseen kuuluu taustoitusta, teknisen toteutuksen laajuus eli scope, ratkaisun hyödyt sekä lisensseihin ja kustannuksiin liittyvää infoa. Olen pohtinut paljon, että missä menee POCin raja ja missä mennään sen yli. Tätä oli myös pohtinut asiakkaan uusi business partner, sillä hän myös tiedusteli aiemmin, että millä tavalla liiketoiminnat ovat POC-hankkeen tekemisestä tietoisia ja ovatko he olleet mukana kehityksessä.

POC-hanke käynnistettiin aiemman business partnerin toimesta, mutta hänen siirryttyänsä uudelle työnantajalle POCin edistäminen on ollut paljolti kehitystiimimme, tarkemmin minun ja ratkaisuarkkitehdin harteilla. Olemme ratkaisuarkkitehdin kanssa toimineet pitkään asiakkaan kanssa sekä asiakkaan että toimittajan roolissa työskennellen, joten sen vuoksi meillä varmaan on ollut melko

vapaat kädet POC-hankkeen edistämiseen, koska asiakas luottaa osaamiseemme, liiketoimintaymmärrykseemme ja arviointikykyymme. Asiakas on lähinnä seurantapalavereissa linjannut etenemisestä, mutta muuten olemme saaneet edistää sitä parhaaksi katsomallamme tavalla. Olisi kuitenkin hankala todistaa uuden sovelluksen toimivan, jos siitä ei olisi tehty jonkinlaista alustavaa versiota. Toisaalta POCin tavoite on todistaa konseptin toimivuus, eikä POC usein ole vielä esitettävissä oleva tuote. (Malsam 2021.) POCin tunnusmerkkejä ovat selkeästi määritelty onnistumiskriteeri, dokumentaatio suorittamisesta sekä ehdotus, miten tulisi jatkaa, jos POC osoittautuu onnistuneeksi (Pratt 2020). Meillä edellä mainitut ovat osittain selviä ja osittain epäselviä. Esimerkiksi dokumentaatio on tällä hetkellä mielestäni hankkeen vaiheen huomioiden hyvällä tasolla. Sen sijaan esimerkiksi itselleni ei ainakaan ole aivan kirkkaana mielessä mitkä ovat POCin onnistumiskriteerit ja kuinka kauan sekä millä budjetilla POCia aiotaan tehdä ennen kunnolliseksi projektiksi kääntämistä. (Malsam 2021.) Mielestäni kehittämämme toteutus vastaisi enemmän prototyyppiä tai MVP:tä (minium viable product) kuin POCia, sillä tekninen toteutus on jo pidemmällä ja meillä on jo osittain toimiva sovellus, jonka avulla pystyy pian suoriutumaan vähimmäisvaatimuksista. (Muzyka 2022.)

POCiin liittyen asiakas oli pyytänyt rajoittamaan Dataversen *person_employee* tauluun vietävien asiakkaan työntekijöiden henkilötietojen viemistä. He olivat huolissaan siitä, että tauluun viedään henkilötietoja kuten etunimi, sukunimi, sähköposti ja puhelinnumero. Tiedot oli alun perin viety testauksen helpottamiseksi, mutta jo aiemmin hankkeen aikana oli puhuttu, että tietoja täytyy jossain vaiheessa rajoittaa, sillä osa tiedoista on turhia. Pyrimme rajoittamaan tietoja GDPR:n vastuullisuusperiaatteen mukaan eli viemällä sinne ainoastaan tarvittavia tietoja. Sähköposti on tarpeellinen, sillä sitä käytämme käyttäjän yhdistämiseen Power Apps:n puolella. Sen sijaan nimitiedot ja puhelinnumero eivät ole pakollisia ja ne poistimme ERP-järjestelmän IAL-näkymästä, josta integraatio käy tiedot poimimassa. (Wolford s.a.)

Sovelluksen avautumiseen liittyvän suorituskykytestauksen osalta lähdin liikkeelle Polyan metodia soveltaen ja kirjasin vaiheet ensiksi tarkasti ylös OneNoteen. Ensimmäisessä vaiheessa ongelma pyritään tunnistamaan. Tässä tapauksessa se tarkoitti muun muassa sovelluksen hitaaseen avautumiseen vaikuttavien tekijöiden tunnistamista, aiheeseen liittyvien termien ymmärtämistä sekä sovelluksen suorituskykyä optimoivien toimenpiteiden kartoittamista. (Kanna 2021.) Microsoft pyrkii Power Appsissa siirtymään imperatiivisesta deklaratiiiviseen kehittämiseen. Deklaratiivinen tarkoittaa, ettei ohjelmalle tarvitse kertoa, milloin tai miten tehtävä tulisi suorittaa, se vaan tulee suorittaa. Imperatiivisessa tarvitaan myös tehtävän suoritukseen vaadittavat vaiheet. (Lindhorst 2021a.)

Microsoft on tuonut deklaratiiivisen suuntauksen myötä sovellusobjektiin uudet ominaisuudet StartScreen ja Formulas. StartScreen-ominaisuuden avulla voidaan yksinkertaisesti määrittellä

mikä näyttö näytetään ensiksi. Tämä sekä yksinkertaistaa että nopeuttaa sovellusta, sillä jatkossa ensimmäisen näytön näyttämisen ei tarvitse odottaa koko OnStart-ominaisuuden valmistumista, vaan näyttö voidaan näyttää heti sovelluksen käynnistyessä. Formulas tarjoaa Excelistä tutun ominaisuuden Power Appsiin, jonka avulla ainakin osa OnStart-ominaisuuden Set- ja Collect-funktiosta voidaan korvata. Formulas-ominaisuuden etuja ovat muun muassa se, että arvo on aina saatavilla ja ajan tasalla. OnStart-ominaisuuden sisällä kaikki muuttujat luodaan, käytetään niitä tai ei. Formulas-ominaisuus tuo OnStart-ominaisuuteen verrattuna mahdollisuuden kaavan luoman muuttujan käyttämisen ainoastaan, kun sitä tarvitaan. (Lindhorst 2021b.)

Seuraavassa eli suunnitelman luomisvaiheessa lähdin pohtimaan testausmetodia, testin rakennetta, toimenpiteitä, jotka tulee suorittaa ennen ja jälkeen yhden testitapauksen sekä varautumista ongelmatilanteisiin. Olin aiemmin tutustunut Power Appsin monitoriin ja tiesin ettei se valitettavasti tue Canvas-sovelluksen käyttämistä mobiililaitteella, joten päädyin valitsemaan ajanottometodiksi ihan perinteisen sekuntikellolla kelloitetun ajan (Microsoft 2022d). Arvelin sen myös olevan riittävä POC-vaiheen tarkoituksiin eikä tässä vaiheessa ollut tarvetta mennä sen tarkemmalle tasolle. Yritin tutkia googlaamalla, että mikä olisi sopiva testimäärä luotettavan testaustuloksen saamiseksi mutten löytänyt mielestäni sopivaa lähdeettä siihen, joten päädyin valitsemaan aiemman testauskokemukseni perusteella määräksi viisi testikertaa.

Ennen testausta huolehdin toimivan version talteen ottamisesta ZIP-tiedostoviennillä, jotta tarvittaessa voisin palauttaa tilanteen ennalleen, jos testauksen yhteydessä sovelluksessa menee jotakin pahemmin rikki. Lisäksi, jotta pystyisin arvioimaan testituloksia, oli suoritettava niin sanottu nollassa testi eli testi, joka kertoo tässä tapauksessa sovelluksen nykyisen nopeuden.

Kuvassa 5 näkyy käyttämäni testitapaukset. Valitsin erilaisia vaihtoehtoja sovelluksen avaamiseen, joita olivat sovelluksen avaaminen mobiilisovelluksen kautta puhelimella, sovelluksen avaaminen selaimen kautta puhelimella ja sovelluksen avaaminen tietokoneella selaimella. Ensimmäisen testikierroksen jälkeen kävi kuitenkin nopeasti selväksi, ettei sovellusta hieman vastoin odotuksia kannata käyttää Microsoft Edge -selaimella. Seuraavan testikierroksen jälkeen pystyin eliminoimaan myös Google Chrome -selaimella käytön mobiililaitteella, sillä sovelluksen avaaminen oli huomattavasti kätevämpää siihen dedikoidun Power Apps launcher -mobiilisovelluksen avulla.

Testin tiedot							
Nollatesti	Testi 1	Testi 2	Testi 3	Testi 4	Testi 5	Välimuistin tyhjennys	Testi 6
- Lähtötilanteessa kaksi työtilausta ja yksi työaikaleimaus							
Sovelluksen avaaminen Chrome-selaimella tietokoneella	7,75	4,33	4,74	4,85	4,52	X	4,60
Testitapaus 1: Power Appsin avaaminen Power Apps launcher-sovelluksen kautta	15,00	10,66	8,21	7,69	7,44	X	10,43
Testitapaus 2: Power Appsin avaaminen puhelimen etusivun pikakuvakkeesta	9,76	10,03	7,08	6,54	7,15	X	10,39
Testitapaus 3: Power Appsin avaaminen Chrome-selaimella puhelimella	7,71	8,52	8,21	8,32	8,41	X	7,44
Testitapaus 4: Power Appsin avaaminen Edge-selaimella puhelimella	25,89	12,41	13,23	13,88	12,62	X	F
Muutos 1	Testi 1	Testi 2	Testi 3	Testi 4	Testi 5	Välimuistin tyhjennys	Testi 6
- Osa OnStart ominaisuuden funktioista Concurrent-funktion sisään							
Sovelluksen avaaminen Chrome-selaimella tietokoneella	4,87	4,63	4,92	4,36	4,69	X	4,61
Testitapaus 1: Power Appsin avaaminen Power Apps launcher-sovelluksen kautta	15,02	6,93	6,54	6,43	6,77	X	13,29
Testitapaus 2: Power Appsin avaaminen puhelimen etusivun pikakuvakkeesta	13,22	6,63	6,27	6,34	6,16	X	12,87
Testitapaus 3: Power Appsin avaaminen Chrome-selaimella puhelimella	9,85	7,62	7,13	6,57	7,67	X	9,49
Muutos 2	Testi 1	Testi 2	Testi 3	Testi 4	Testi 5	Välimuistin tyhjennys	Testi 6
- Lisää funktioita Concurrent-funktion sisään							
Sovelluksen avaaminen Chrome-selaimella tietokoneella	4,63	4,45	4,38	4,43	4,49	X	4,47
Testitapaus 1: Power Appsin avaaminen Power Apps launcher-sovelluksen kautta	11,85	7,01	6,68	5,94	6,65	X	11,94
Testitapaus 2: Power Appsin avaaminen puhelimen etusivun pikakuvakkeesta	-	-	-	-	-	-	-
Muutos 3	Testi 1	Testi 2	Testi 3	Testi 4	Testi 5	Välimuistin tyhjennys	Testi 6
- App Checkerin mukaisia korjauksia sekä värit globaaleina muuttujina erillisten kontrollien sijaan							
Sovelluksen avaaminen Chrome-selaimella tietokoneella	4,65	4,25	4,27	4,38	4,31	X	4,60
Testitapaus 1: Power Appsin avaaminen Power Apps launcher-sovelluksen kautta	8,59	4,97	4,73	4,38	4,41	X	9,34
Testitapaus 2: Power Appsin avaaminen puhelimen etusivun pikakuvakkeesta	8,03	5,30	4,70	4,11	4,31	X	8,12

Kuva 5. Suorituskykytestien tiedot (Microsoft Excel 2022a)

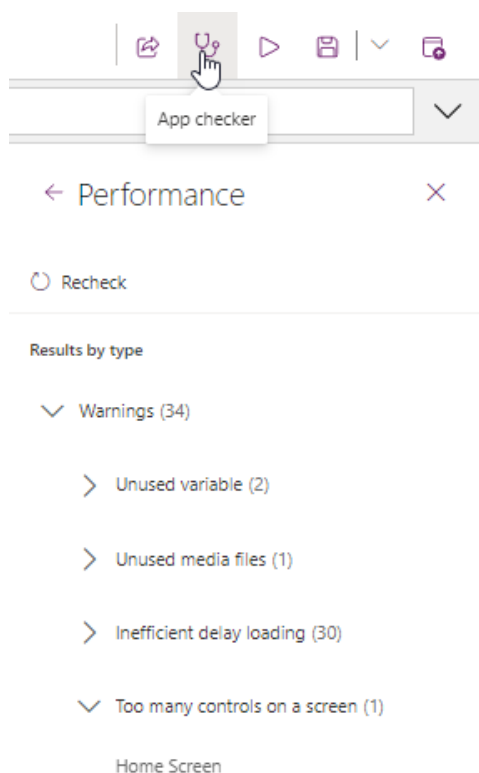
Kuten kuvasta 6 käy ilmi, suorituskykytesteistä pystyi päättämään, että sovelluksen sulkeminen tai välimuistin tyhjentäminen aiheuttavat notkahduksen sovelluksen avautumisen nopeuteen. Huomasin lisäksi, että jokainen valmiiksi tutkimani ja määrittelemä muutos aiheutti positiivisen vaikutuksen suorituskykyyn ja se näkyi etenkin mobiiliin osalta, jossa lopputuloksena nopeus lähes puollittui. Selaimen nopeuksissa näkyi myös muutosta mutta muutos oli huomattavasti maltillisempi ja tapahtui saman sekuntiyksikön sisällä.

Tilastot			
Keskiarvo ilman välimuistin tyhjennyksen aiheuttamaa hidastusta	Keskiarvo	Min	Max
4,61	5,13	4,33	7,75
8,50	9,91	7,44	15,00
7,70	8,49	6,54	10,39
8,37	8,10	7,44	8,52
13,04	15,61	12,41	25,89
Keskiarvo ilman välimuistin tyhjennyksen aiheuttamaa hidastusta	Keskiarvo	Min	Max
4,65	4,68	4,36	4,92
6,67	9,16	6,43	15,02
6,35	8,58	6,16	13,22
7,25	8,06	6,57	9,85
Keskiarvo ilman välimuistin tyhjennyksen aiheuttamaa hidastusta	Keskiarvo	Min	Max
4,44	4,48	4,38	4,63
6,57	8,35	5,94	11,94
-	-	-	-
Keskiarvo ilman välimuistin tyhjennyksen aiheuttamaa hidastusta	Keskiarvo	Min	Max
4,30	4,41	4,25	4,65
4,62	6,07	4,38	9,34
4,61	5,76	4,11	8,12

Kuva 6. Suorituskykytestien tilastot (Microsoft Excel 2022b)

Power Apps kehityksessä tulee väistämättä vastaan virheitä ja varoituksia. Olen huomannut, että ne tulevat usein vastaan funktioita luodessa kaavarivin kautta. Silloin Power Apps näyttää kehitystudioon käyttöliittymässä punaisen virhekuvakkeen tai keltaisen varoituskolmion, jonka avulla ongelmaan pääsee helposti käsiksi.

Huomasin suorituskyvyn parantamista tehdessäni, että sovelluksesta löytyy myös ”Sovelluksen tarkistus” -ominaisuus (kuva 7). Se näyttää kaikki kohdat, joista sovelluksessa löytyy virheitä ja varoituksia. Jokaisen virheen tai varoituksen kohdalta pääsee tarkemmin tutkimaan sen yksityiskohtia. Tarkemmista tiedoista voi selvittää muun muassa mikä virheen tai varoituksen aiheuttaa ja saada valmiita suosituksia mitä sille voisi tehdä. Huomasin sovelluksen tarkistuksen kautta esimerkiksi, että jos yhdellä näytöllä on liikaa kontrolleja (yli 300), niin sovelluksen tarkistus huomauttaa suorituskyvyn mahdollisesta heikentymisestä. (Delano 2018.)



Kuva 7. Power Apps sovelluksen App Checker (Microsoft Power Apps 2022a)

Sovelluksen tarkistus on jatkossakin erittäin hyödyllinen työkalu, jos sovelluksen suorituskykyä tarvitsee tarkistaa ja muutenkin aion jatkossa huomioida heti kehitysvaiheessa sovelluksen ilmoittamat varoitukset ja kohdat, joissa on mahdollista pienillä toimenpiteillä optimoida, jotteivät ne pääse liikaa kasautumaan.

HR Häirintäilmoituksen tuotantoon vientiin liittyen selvitettävät asiat saimme tällä viikolla selvitettyä ja nyt olisi mahdollisuus alkaa siirtämään sovellusta tuotantoon. Tähän muodostui kuitenkin vielä käyttöoikeuksiin liittyvä este, koska HR:llä heräsi huoli yleisten järjestelmävalvojen oikeuksista nähdä HR Häirintäilmoitukseen liittyvää sensitiivistä dataa, ja lähtökohtaisesti he olisivat halunneet, ettei kukaan muu kuin HR näe kyseistä dataa. Yleisillä järjestelmävalvojilla on pääsy kaikkeen Microsoft 365 ympäristön sisällä, joten heiltä datan piilottaminen on käytännössä mahdotonta. Tutustuin asiaan tutkimalla Googlen keskustelufoorumeja, Microsoftin omia ohjeita sekä keskustelemalla ulkopuolisen Power Apps konsultin kanssa. Ymmärsin, että joidenkin tietolähteiden osalta voisi olla mahdollista rajoittaa yleisten järjestelmävalvojen pääsyä ottamalla sovellukselle määritetty ryhmä heiltä pois, mutta järjestelmävalvojina he kuitenkin voisivat lisätä oikeuden takaisin. Kysymys kuuluukin nyt, että kenellä kaikilla tällä hetkellä on yleiset järjestelmävalvojan oikeudet ja tarvitsevatko kaikki kyseisiä oikeuksia? Toisaalta myös järjestelmävalvojiin pitäisi pystyä luottamaan, joten onko HR:n vaatimus oikeuksien poistamisesta kuitenkin oikea?

Tietoturvan näkökulmasta ympäristön yleisten järjestelmänvalvojien määrä tulisi pitää maltillisena. Suosituksen mukaan yleisiä järjestelmänvalvoja olisi hyvä olla kahdesta neljään tunnusta. (Microsoft 2022e.) Tällä varmistetaan, ettei oikeuksia jaeta liian laajalle joukolle ja jaettujen käyttäjien osalta pystytään tunnistamaan henkilöt varmasti oikeiksi henkilöiksi esimerkiksi MFA:n avulla sekä auditoimaan heidän tekemiään toimintoja ja mahdollisesti sitä kautta puuttumaan väärinkäytöksiin. (Bradley 2020.)

Tutkin asiakkaan ympäristössä olevien yleisten järjestelmävalvojien määrää ja huomasin niitä olevan reilusti yli Microsoftin suosituksen. En heti keksinyt syytä miksi yleisiä järjestelmänvalvojan oikeuksia oli jaettu niin paljon ja olin yhteyksissä asiakkaan ICT:hen kysyäkseni asiasta. Ymmärrän myös HR:n huolen asiasta mutten konsultin roolissa oikein voi paljon asialle tehdä. Sain asiakkaan ICT-osastolta ohjeeksi olla yhteydessä heidän tietoturvapääällikkönsä, jolta voisi saada lausunnon asiaan. Jotta pystyn pysymään tuotantoon viennin aikataulussa, tämä asia tulisi selvittää pikimmiten. Aion olla yhteydessä tietoturvapääällikköön pian, ellei HR:ltä tule jotain ratkaisua asiaan.

3.5 Seurantaviikko 5 (vko 39)

Maanantai 26.9.2022

Tavoite: Käydä läpi asiakkaan ICT:n kanssa liiketoimintademon materiaali eli esittää heille demo ja kysyä kommentteja. Tämän lisäksi muun muassa uuden työtilauksen integraatio kaipaa vielä työstämistä.

Toteutuminen: Päivä oli kiireinen muiden tuotantokiireiden kanssa. Kävimme kuitenkin aamupäivästä pariin otteeseen liiketoimintademoa läpi eli sparrasimme ICT:n sekä toimintamallipäällikön kanssa ja saimme hyvää palautetta esityksestä. Viime viikolla uuden työtilauksen integraation haasteet työn alle ottanut integraatiokehittäjä oli poissa tänään, joten kehitystä ei pystytty jatkamaan.

Tiistai 27.9.2022

Tavoite: Saada uuden työtilauksen siirtämisen integraatio toimimaan, jotta liiketoimintademossa voimme demota työtilauksen perustamisen.

Toteutuminen: Kävimme asiaa läpi integraatiokehittäjän kanssa sillä toinen uusista integraatioasi-antuntijoista oli sairastunut eikä ollut sen vuoksi pystynyt korjaamaan integraatiota. Totesimme ratkaisuarkkitehdin kanssa yhteen ääneen, että nyt yritetään saada integraatio toimimaan erillisenä integraationa, jolloin integraatiosta tulee huomattavasti selkeämpi ja helpompi hahmottaa. Integraatiota korjattaessamme törmäsimme virheeseen: "Resource not found" ja ihmettelimme mistä se

voi johtua. Kävi ilmi, että Azuren päähän konfiguroitu liitin (connector) käytti porttia 53080 ja SOAP-rajapinta ERP-järjestelmän puolella käytti porttia 54080. Integraatiokehittäjä sanoi sen olleen alusta asti 53080 eikä siihen ollut aiemmin koskettu. Tämän näkyi myös muissa rakentamisessamme integraatioissa, sillä esimerkiksi myös työtilauksen muutokset kaatuivat samaan virheeseen. Muutimme liittimen (connector) käyttämään oikeaa porttia, jonka jälkeen integraatiot alkoivat jälleen toimimaan. Tästä tuli mieleen, että olikohan viime viikon ongelmat vaan kiinni yhdistimestä ja olisiko add, modify, delete triggeri voinut toimia oikean portin kanssa jo silloin. Pohdimme myös mistä moinen muutos olisi voinut johtua ja epäilimme sen johtuneen sovellustoimittajan testausympäristöön tekemästä päivityksestä.

Keskiviikko 28.9.2022

Tavoite: Tavoitteena viimeistellä liiketoimintademon esitysmateriaali sekä katsoa valmiiksi asiat esitystä varten, jotta esitys sujuu mahdollisimman sujuvasti.

Toteutuminen: Kävimme päivittäispalaverissa tilanteen läpi ja totesimme, että kaikki on nyt tehty liiketoimintademoa varten teknisen toteutuksen osalta ja pyysimme etteivät muut tiimiin jäsenet tee enää muutoksia POC-hankkeeseen. Tällä pyrimme varmistamaan, että sovellus ja taustajärjestelmät varmasti toimivat sulavasti liiketoimintademossa. Ratkaisuarkkitehdin kanssa lisäksi kävimme vielä kertaalleen läpi esitysmateriaalin ja sovimme esityksen osuuksista. Minulle jäi taustoitusta sekä ratkaisun hyödyt ja ratkaisuarkkitehti puolestaan kertoo sovelluksen teknisen toteutuksen laajuudesta sekä Power Apps:n tarjoamista mahdollisuuksista.

Tein vielä illalla joitakin pieniä parannuksia visuaaliseen ilmeeseen sekä avasin valmiiksi ERP-järjestelmästä tarvittavat näkymät ja putsasin Power Apps:n puolelta turhat testeissä tehdyt datat demoa varten.

Muutoin päivä meni suurilta osin muiden opinnäytetyön kontekstiin kuulumattomien työtehtävien parissa.

Torstai 29.9.2022

Tavoite: Päättävänä saada vihreää valoa liiketoimintademon osalta, jotta päästään jatkamaan POC-hankkeen kanssa eteenpäin. Olisi myös hienoa, jos saataisiin jotain suuntaviivoja liiketoiminnalta jatkos osalta.

Toteutuminen: Valmistauduin aamulla vielä esitykseen varmistamalla, että eiliset valmistelut olivat paikallaan. Kytinkin myös esitystä varten uuden työtilauksen ilmoituksen päälle, jotta voin demossa näyttää ilmoituksen, joka tulee, kun uusi keikka on saatavilla.

Meidän lisäksi esityksessä oli mukana paljon asiakkaan edustajia kuten kehitysjohtaja, valmennuspäällikkö, uusi business partner, sovellusarkkitehti, muita päälliköitä, pääkäyttäjiä ja muita asiakkaan ERP-järjestelmän käyttäjiä. Suurin osa paikallaolijoista oli meille entuudestaan tuttuja mutta joukossa oli myös uusia henkilöitä, joten esittäydymme lyhyesti heille.

Esitys koostui PowerPoint-kalvoista ja varsinaisesta sovellusdemosta, jossa näytin sovellusta näyttöä jakamalla sekä puhelimella että tietokoneella.

Perjantai 30.9.2022

Tavoite: Sovimme eilen BAS-tiimille pidettävästä demosta, joka on tarkoitus pitää ensi tiistaina. Sitä varten olisi tavoitteena korjata eilisen liiketoimintademon aikana havaitut ongelmat, testata työnkulut sekä kääntää esitys englanniksi.

Toteutuminen: Integraatiokehittäjät / -asiantuntijat olivat kiireisiä koko päivän asiakkaan useissa laajavaikutteisissa tuotantohaasteissa, joten kävimme vain lyhyesti päivittäispalaverissa eilisen liiketoimintademon jälkeisten tarvittavien asioiden tilanteen.

Käänsin PowerPoint-esityksen englanniksi, jotta voimme sen helposti esittää ensi tiistaina BAS-tiimille. Paljon tärkeämpää oli kuitenkin selvittää, että miksi työtilauksen kohdistaminen epäonnistui, vaikka aiemmin se oli toiminut oikein. Lisäksi uuden työtilauksen luomisen prosessi mobiilisovelluksessa ei ollut vielä valmis, sillä työtilaus oli eri tilassa ERP-järjestelmässä ja Power Appsissa. Piti selvittää, että miten se saadaan samaan tilaan molemmissa järjestelmissä, jotta sitä voi heti mobiilissa luonnin jälkeen käsitellä.

Päätin kokeilla, sillä olin mielestäni tarpeeksi nähnyt ja avustanut integraatiokehittäjää aiempien integraatioiden tekemisessä. Sain tehtyä integraation muutoksen kertaheitolla oikein. Uskon, että ymmärrys kokonaisuudesta auttaa myös integraation kehittämisessä, sillä nyt minun ei tarvinnut kertoa toiselle mitä haluan, minun piti vain saada haluamani toiminnallisuus luotua Logic Appsien puolelle itse.

Viikkoanalyysi 5

Viikon osaamisen kehittäminen liittyi suurimmaksi osaksi torstain liiketoimintademon valmistautumiseen sekä demon pitämiseen. Tekninen toteutus oli tällä viikolla jo hyvässä mallissa ja uuden työtilauksen integraation korjaamisen lisäksi siihen tehtiin ainoastaan joitakin pieniä viilauksia. HR Häirintäilmoituksen osalta viikko ei tuonut mitään uutta ja pallo on edelleen HR:llä. He ovat kieltäneet minua olemasta yhteydessä tietoturvapäällikköön ja sanovat selvittävänsä asiaa tietohallinnon

palvelujohtajan kanssa ensin. Olen hieman hankalassa välikädessä, sillä toisaalta minun pitäisi pystyä edistämään asiaa, jotta pääsisimme pian tuotantoon testaamaan ja varmistamaan sovelluksen toimivuuden. Toisaalta en haluaisi puuttua asiaan liikaa, sillä asia liittyy enimmäkseen asiakkaan sisäisiin prosesseihin enkä halua toimia toimeksiantajani eli HR:n selän takana, sillä siitä voisi aiheutua kitkaa välillemme.

Ennen torstain liiketoimintademoa pohdin demon onnistumisen kriteerejä. Ensinnäkin mielestäni esityksen teknisestä osuudesta liiketoiminnoille täytyy tulla tuntuma, että kyseinen tuote voisi oikeasti toimia korvaajana nykyiselle mobiilisovellukselle. Toisekseen esityksen tietojen pitää tulla lyhyesti, ytimekkäästi sekä asiakasta kiinnostavasti esille. Demoon mentäessä on myös syytä tutustua osallistujiin etukäteen ja ymmärtää heidän roolinsa (Hart 2021). Tapoja, joilla demoon voi valmistautua on monia ja jokainen demo on oma uniikki tapauksensa. Yksi mahdollinen lähestymistapa on kysyä ennakkoon asiakkaalta mitä he haluaisivat esityksen sisältävän. Tein juuri näin ja sain vastaukseksi, että lyhyt esittely sekä tekninen demo riittää.

Huomasin esityksen aikana, että olin keskittynyt nykyisen toteutuksen scopen mukaiseen demoamiseen niin paljon, että olin unohtanut pohtia mahdollisia POC-hankkeen scopen ulkopuolisia kysymyksiä valmiiksi. Näistä ainakin iOS-käyttöön liittyvä kysymys yllätti, sillä tiedän asiakkaan tällä hetkellä käyttävän ainoastaan Android-laitteita. Tästä opin, että on hyvä pitää myös muut mahdollisuudet mielessä eikä vain luottaa tuttuun, turvalliseen ja valmiiksi pohdittuun. (Lifesize 2019.) Vaikka hetken epäröinkin vastata, niin tiesin kuitenkin ennakkoon, että Power Apps toimii sekä Android että iOS-laitteilla. Toinen liiketoimintoja paljon kiinnostava kysymys oli, että saadaanko käyttöönottoa helpotettua. Siinä on pitkään käyttäjillä ollut haasteita ja etenkin vuokratyöntekijöiden kohdalla käyttö on raskasta. Prosessi on nykyisellään haastava, monivaiheinen ja vaatii monenlaisia oikeuksia oikeassa järjestyksessä annettuna sekä itse sovelluksen ensimmäinen käyttöönotto kestää internet-yhteyden nopeuden mukaan 10–20 minuuttia. Power Apps tuo tähän huomattavaa helpotusta, sillä käyttäjä pääsee kirjautumaan Office 365-tilin avulla sovellukseen, jonka jälkeen sovellus on heti käytettävissä. Taustalla vaaditaan ainoastaan Power Apps per app plan-lisenssi sekä sovelluksen jakaminen käyttäjälle. Muiden kysymysten jälkeen asiakkaan pitkän linjan konkarilla, nykyisin valmennuspäällikkönä toimivalla henkilöllä oli enää yksi kysymys: ”Koska tämä saadaan tuotantokäyttöön?” ”Onnistuisiko tuotantotestaus jo heti ensi viikolla?”

Tähän meillä ei tietenkään ollut suoraa vastausta antaa ottaen huomioon POC-vaiheen mutta jostain syystä ratkaisuarkkitehti meni kuitenkin arvaamalla arvioimaan (guesstimate) että kahden viikon päästä voisi olla tuotantokäyttäjille versio testattavissa. Tämä arvio ei perustunut faktatietoon emmekä olleet keskustelleet asiasta yhtään etukäteen vaan arvio oli puhdas arvaus. Ratkaisuar-

kitehti arveli varmaan, että asiakas odottaa vastausta heti, mutta tilanteessa olisi tarkan arvion sijaan kannattanut antaa jonkinlainen suuntaa antava väli (confidence interval), sillä ihmiset tuppavat olemaan ylioptimistisiä omissa arvioissaan ja aika-arvioihin vaikuttavat monet seikat kuten resurssien saatavuus, muut työtehtävät sekä mahdolliset vastoinkäymiset. (Marchewka 2014, 162.)

Yksi mahdollisuus olisi ollut ehdottaa palaveria, jossa olisi asiakkaan kanssa sovittu jatkosta ja käytetty esimerkiksi bottom-up arviointitekniikkaa, jossa olisimme voineet palastella tekemisen pienempiin osiin ja arvioida yksittäisten osioiden kestoa tarkemmin. Esimerkkejä osioista voisivat olla tarvittavien toiminnallisuuksien kartoittaminen, käyttöliittymän tarkempi suunnittelu sekä sovelluksen nimeäminen ja brändäys. Toiminnallisuudet voisi edelleen jakaa pienempiin osiin, joista esimerkkeinä kulujen kirjaus, työaikakirjausten tekemisen muokkaaminen, niin että myös toisen palvelulinjan työaikakirjaukset saadaan kirjattua, mobiiliostot ja työtilausten ketjutus. (Marchewka 2014, 164.)

Tekninen demo meni suurimmaksi osaksi suunnitellun mukaisesti. Sain demottaa työaikakirjausten ja työtilausten välisen logiikan, työaikakirjauksen tekemisen ja niiden lähes reaaliaikaisen siirtymisen ERP-järjestelmään. Kerroin sovelluksen hyödyntämästä kielikäännös Excelistä, jota asiakas voi tarpeen mukaan muokata. Onnistuin demoamaan myös uuden työtilauksen luomisen mobiilissa. Usein esityksiä pidettäessä voidaan pelätä demoeefektiä, jolla tarkoitetaan tilannetta, jossa sovellus ei toimikaan kuten sen kuuluisi tai kuten se testatessa on aiemmin toiminut. Kaikesta huolellisesta valmistautumisesta huolimatta niin kävi myös demossamme. Työtilauksen käsittely ei jostain syystä toiminut toivotulla tavalla ensimmäisen kerran demotessa. Tein työtilauksen käsittelyä sovelluksessa ja olin näyttämässä asiakkaalle ERP-järjestelmästä, että tänne kirjaukset tulevat samaan tapaan kuin ne tulevat nykyisestä mobiilisovelluksesta. Harmikseni huomasin, että työtilauksen tila ei ollut muuttunut mihinkään. Tarkistin nopeasti toiselta näytöltä mitä Dataversen tietokanta sekä ajot näyttivät taustalla ja löysin sieltä virheen "Equipment functional object does not exist".

Pelkäsin hieman, että asiakas tarttuisi tuohon päivittymättömyyteen mutta huomasin, ettei se juurikaan kiinnostanut asiakasta. Ehkä he luottivat sanaamme ja ymmärsivät että tässä vaiheessa kehitystä voi esiintyä tuollaisia vikoja. Pohdimme jälkepäin ratkaisuarkkitehdin kanssa, että olisivat kohan edes huomanneet virhettä ellen olisi siitä erikseen maininnut. Omiin peruseriaatteisiin kuuluu kuitenkin rehellisyys ja oli itsestään selvää, ettei sitä yritetty lähteä peittelemään mitenkään. Läpinäkyvyys ja rehellisyys lisäävät asiakkaan luottamusta tekemiseemme parantaen ja lujittaen yhteistyötä. Asiakkaalla on nykyään paljon vaihtoehtoja ja varmasti epäluotettava tekijä vaihtuisi nopeasti toiseen. (Robinson 2020.)

Esityksen jälkeen ihmettelin kuitenkin, kuinka juuri tuo ominaisuus, jota oli hiottu pitkään ja hartaasti ja joka oli yksi koko sovelluksen toiminnan kulmakivistä, ei sitten kuitenkaan h-hetkellä enää

toiminutkaan. Asiaa tutkittuamme selvisi, että ADF:n integraatio, joka siirtää työtilaukset ERP-järjestelmästä Power Appsiin oli jotenkin mennyt rikki. Työtilaukset kyllä siirtyivät Power Appsiin, mutta ne tulivat ilman työtilausnumeroa. Käytin tässä nimenomaisessa epäonnistuneessa tapauksessa työtilausta, jonka tein ERP-järjestelmässä ja joka siirtyi sieltä ADF:n kautta Power Appsiin. Epäilimme että viimeaikaisin kehitys uuteen työtilausintegraatioon oli jotenkin sotkenut ADF:n toimintaa, sillä uudet työtilaukset eivät toimineet, mutta vanhoilla työtilauksilla pystyin kuitenkin lopulta näyttämään työtilauksen käsittelyn toimintaa.

Business partner nosti esityksen jälkeen esille myös sovelluksen nimeämisen ja brändäyksen. Olimme työstönimenä kutsuneet sovellusta samalla nimellä, kun loppukäyttäjät nykyisen sovelluksen tuntevat mutta totesimme ettei kannata käyttää samaa nimeä. Vanhasta nimestä voi tulla huonoja assosiaatioita uuden ja vanhan sovelluksen välillä. Tärkeää olisi pystyä brändäämään sovellus täysin erillisenä, uutena, modernina ja parempana kuin nykyinen. Business partner ehdotti, että nimestä voisi järjestää nimikilpailun tai kehitystiimi voisi kehittää sille paremman nimen. Eräs ehdotus oli, että nimikilpailu järjestetään Yammer-viestintäkanavan kautta. Se oli kaikkien mielestä hyvä idea, sillä silloin saadaan pienellä vaivalla kaikki halukkaat mukaan luomaan sovellukselle omaa tarinaa, aloittamalla sovelluksen nimestä. (Norambuena 2020.)

Olimme myös esityksen aikana puhuneet, että kirjoittaisimme Yammeriin viestin loppukäyttäjille meneillään olevasta kehityksestä. Asiakkaan viestinnästä vastaava henkilö oli sitä mieltä, että jo kehitysvaiheessa mainostaminen voisi innostaa loppukäyttäjiä. Tämä varsinkin sen vuoksi, että vanha sovellus on aiheuttanut harmia ja negatiivista palautetta kentältä jo vuosia. Yleisesti sovelluksesta työntekijöille puhuminen voisi myös auttaa sovelluksen parempaan vastaanottoon. Tähän toki tarvitaan hyvä ja onnistunut demo myös loppukäyttäjille, jotka pääsevät ensimmäisinä testaamaan sovelluksen käyttöä ja sen jälkeen muodostamaan oman mielipiteensä sovelluksesta. Sillä on suuri merkitys, koska sen jälkeen puskaradio lähtee varmasti kiertämään ja muutkin loppukäyttäjät kuulevat onko sovellus hyvä vai ei. (Ray 2015.)

Lähetin iltapäivästä esityksen jälkeen niin kutsutun kiitosviestin, jossa kiitin kutsusta heidän kehitystiiminsä palaveriin, jaoin esitysmateriaalin, kertosin alustavat, esityksen aikana kasatut seuraavat vaiheet sekä lopetin viestin mahdollisuuden olla matalalla kynnyksellä yhteydessä, mikäli heillä tulee kysyttävää (Walnut 2021). Jäimme ratkaisuarkkitehdin kanssa pohtimaan, että miten POC-hankkeen kanssa edetään, nyt kun liiketoiminnot osoittivat selkeää kiinnostusta siihen mutta mitään tarkempaa suunnitelmaa ei vielä ole. Tuntui, että tilanne jäi esityksen loputtua vähän leijumaan eikä kukaan oikein suuremmin ottanut asiasta koppia. Esimerkiksi olisin ehkä odottanut, että business partner olisi hieman summeerannut tilannetta esityksen loppuksi ja avannut keskustelua

jatkon osalta. Jotenkin hanakala nähdä, että me ratkaisuarkkitehdin kanssa ulkopuolisina konsultteina voisimme tehdä päätöksen asian edistämisen suhteen. Mielestäni olisi erittäin tärkeää vähintäänkin pystyä määrittelemään tarvittavat toiminnallisuudet sekä käyttöliittymän ulkoasu pääpiirtein. Joka tapauksessa kiitosviestillä pyrin jatkamaan keskustelua liiketoimintojen kanssa ja tarjoamaan jonkinlaisen ratkaisuehdotuksen etenemisestä. Nyt pallo on liiketoiminnoilla ja asiakkaan ICT:llä mutta on tärkeää pysyä aktiivisena myös itse.

Näin jälkepäin ajateltuna, olisin voinut lukea paljon enemmän sovelluksen markkinointiin liittyvän liiketoimintademon pitämisestä ja hyödyntää niistä saatuja oppeja demossa. Vaikka olenkin paljon koulutuksia ja demoja pitänyt, oli tämä kuitenkin ensimmäinen kerta, kun pidin demon kehittämistäni sovelluksesta. Demon laadun ja johdonmukaisuuden parantamiseen auttaa tutkiminen, huolellinen suunnittelu ja harjoittelu. Meidän oli tavoitteena myös pitää demo ensin ”sisäisesti” asiakkaan kumppaniyrityksen asiantuntijoiden kesken mutta se jäi nyt kuitenkin aikatauluhaasteiden vuoksi pitämättä. (Lowery 2020.) Demon laatu on tärkeässä roolissa, sillä hyväkin tuote tai palvelu voidaan pilata huonolla suunnitellulla ja huonosti toteutetulla demolla (Kothadiya s.a.). Demoista ei pitäisi tehdä myöskään liian pitkiä ja heti aluksi pitäisi siirtyä tuloksiin sen sijaan, että kasaisi jännitystä sillä siinä voi menettää asiakkaan kiinnostuksen (Berloty 2021). En huomannut vastaavaa esityksemme osalta, vaikka se lopulta kestitkin yli tunnin. Jotenkin vaikutti siltä, että asiakas oli todella kiinnostunut esityksestä sekä käsillä olevista mahdollisuuksista ja kiinnostus vaikutti säilyvän läpi koko demon.

Huomasin laskutusta tehdessäni, että minulta puuttui tuntikirjauksia useilta päiviltä ja totesin, että on hankala muistaa ja löytää tarkempia tietoja edes muutaman päivän takaa, jos niitä ei ole mihinkään kirjannut. Aion jatkossa hyödyntää työnohjausjärjestelmämme kellotustoiminnallisuutta, jonka avulla voin työtä aloittaessani kirjata lyhyesti mitä olen tekemässä ja laittaa kellotuksen käyntiin. Se vaatii jonkin verran rutiinin opettelemista sekä itsekuria, sillä tehtävästä toiseen hyppäämisen yhteydessä pitää muistaa aina lopettaa kellotus ja aloittaa se seuraavan tehtävän kohdalla uudestaan. Kuitenkin siitä olisi huomattavaa apua laskutettavien tuntien osalta, kun ei tarvitsisi kaivaa sähköpostin, kalenterin ja Teams-keskusteluhistorian avulla, että mitä on tehnyt. Samalla kellottaminen tuo läpinäkyvyyttä laskutuksen osalta sekä itselleni, asiakkaalle että myös yrityksellemme, koska tähän asti olen kirjannut molemmista Power Apps tehtävistä lähinnä tunnit tehtävää kohden mutten tarkemmin mitä niihin kuuluu. (Projector 2022.)

3.6 Seurantaviikko 6 (vko 40)

Maanantai 3.10.2022

Tavoite: Korjata demossa esiintynyt ongelma sekä käydä esityksen käänös yhdessä ratkaisuarkkitehdin kanssa, jotta voimme huomenna demota ”sisäisesti” englanninkieliselle yleisölle. Päivän päivittäispalaverissa tavoitteena tarkistaa mikä tilanne työn alla olevien tehtävien osalta on.

Toteutuminen: Päivä oli tuottoisa ja saimme paljon aikaiseksi. Kävimme päivittäispalaverissa läpi integraatiokehittäjän ja muiden tiimiläisten kanssa demossa esiintynyttä ongelmaa ja sovimme sen korjauksesta. Heillä vaikutti olevan tuotantohaasteiden kanssa kiire enkä ollut aivan varma olivatko he täysin ymmärtäneet ongelmaa, joten ehdotin, että voisin katsoa sitä ensin itse.

Sain ongelman korjattua korjaamalla aiemmin luotua uuden työtilauksen integraation triggeriä. Kyseinen integraatio aiheutti ongelman triggeröitymällä samaan aikaan kun ADF siirsi uuden työtilauksen asiakkaan ERP-järjestelmästä Power Appsiin. Tämä sekoitti Power Appsin tietoja ja aiheutti demossa esiintyneen demoefektin työtilauksen käsittelyyn. Lisäsin myös käsittelyn, jolla tarkistetaan pitääkö käyttäjä luodun työtilauksen vai siirtääkö hän työtilauksen pääjärjestelmään toiselle tekijälle kohdistettavaksi. Muutin uuden työtilauksen integraation triggeröitymään ainoastaan silloin kun *original_system_id* kentässä on arvo. Triggeröitymisen jälkeen integraatio tarkistaa onko työtilauksella arvona pidä vai palauta. Sen mukaan työnkulku jatkaa eteenpäin ja suorittaa joko työtilauksen päivittämisen pitämistä varten tai siirtää työtilauksen pääjärjestelmään ja poistaa sen samalla tietokannasta.

Saimme luotua dynaamisen mäppäystiedoston ADF:ään, joka helpottaa sarakkeiden lisäämistä jatkossa. Aiemmin lisäykset aiheuttivat tarpeen mäpätä kaikki sarakkeet uudestaan mutta nyt ne voi tehdä dynaamisesti päivittämällä JSON-mäppäystiedoston. Mäppäystiedostoon sarakkeelle tarvitsee lisätä source eli lähdetietokannan sarake sekä sink eli vastaanottavan tietokannan sarake. Tämän lisäksi ADF:ssä täytyy muokata kyselyä, joka hakee lähdejärjestelmästä tiedot, jotta se osaa tuoda myös uuden kentän tiedot oikein.

Parantelin myös kuvassa 8 näkyvää, aiemmin luomaani lokalisaatio Exceliä, jotta sitä on helpompi lukea ja siihen on jatkossa helpompi tehdä lisäyksiä. Paranneltuna se on myös helpompi esitellä ja opastaa asiakkaalle. Lisäsin muun muassa Exceliin näytön nimen sekä tekstin tarkemman sijainnin sovelluksen rakenteessa. Näin ollen on helpompi nähdä missä paikassa kyseistä käännöstä käytetään tai kääntäen löytää esimerkiksi kaikki käännökset, jotka ovat kotinäytöllä.

	A	B	C	D	E	F	G
1	TextIDLanguageTag	TextID	LanguageTag	LocalizedText	Screen	Detailed location / container	Notes
2	HomeScreenTitleTab1en	HomeScreenTitleTab1	en	Workday entries	Home Screen	Home Tabs Container	
3	HomeScreenTitleTab2en	HomeScreenTitleTab2	en	Work orders	Home Screen	Home Tabs Container	
4	HomeScreenTitleTab3en	HomeScreenTitleTab3	en	New work order	Home Screen	Home Tabs Container	
5	HomeScreenTitleTab4en	HomeScreenTitleTab4	en	Objects	Home Screen	Home Tabs Container	
6	HomeScreenTitleTab5en	HomeScreenTitleTab5	en	Settings	Home Screen	Home Tabs Container	
7	btnStartWorkingTimeen	btnStartWorkingTime	en	Start worktime	Home Screen	Home Tabs Container	
8	btnStartLunchen	btnStartLunch	en	Start lunch	Home Screen	Work Day Entry Container	
9	btnEndLunchen	btnEndLunch	en	End lunch	Home Screen	Work Day Entry Container	
10	btnEndWorkingTimeen	btnEndWorkingTime	en	End worktime	Home Screen	Work Day Entry Container	
11	headerWeekDayen	headerWeekDay	en	Weekday	Home Screen	Work Day Entry Container	

Kuva 8. Excel lokalisointitaulukko (Microsoft Excel 2022c)

Muutoksessa oli tärkeää huomata, että kategorisointi ei voinut olla lokalisointitaulukossa vaan sen piti olla Excelissä erillisinä sarakkeina joko ennen taulukkoa tai taulukon jälkeen. Tiedoston lataaminen Power Appsiin siten, että sarakkeet olivat lisätty Power Appsin hyödyntämään lokalisointitaulukkoon aiheutti Power Appsin puolella virheen. Sen vuoksi niiden piti olla erillään taulukosta. (Microsoft 2022g.)

Tiistai 4.10.2022

Tavoite: Viimeistellä englanninkieliseen demoon tarvittavat asiat

Toteutuminen: Kävimme päivittäispalaverissa tilannetta läpi ja kaikki näytti hyvältä demoa varten. Sovimme, että tarkistamme ratkaisuarkkitehdin kanssa viime perjantaina tekemäni käännökset ja viimeistelemme englanninkielisen demon.

Yrityksemme toimitusjohtajan varakäsi, toiminnanjohtaja oli minuun yhteydessä ja kyseli, että olenko huomannut asiakkaalla tarvetta UI/UX-suunnittelulle tai onko kukaan semmoisen perään kysellyt. Tämä kysely johtui siitä, että eräs yrityksemme kollega on tulossa takaisin äitiyslomalta ja toiminnanjohtaja yritti selvittää, että mitä hommia hänelle löytyisi. En ollut muuten tällaista huomannut tai kuullut asiakkaan suunnalta, mutta minulla juolahti mieleen, että voisinkohan mahdollisesti hyödyntää UI/UX-suunnittelijan osaamista esimerkiksi POC-hankkeessa tai mahdollisissa tulevilla hankkeissa. Keskustelimme asiasta tovin ja sovimme, että kysyn asiakkaalta jossain sopivassa välissä, että miltä kuulostaisi, jos meiltä UI/UX-suunnittelija pohtisi nykyistä toteutusta UI/UX-kulmalla.

Keskiviikko 5.10.2022

Lyhyen päivittäispalaverin lisäksi ei muita opinnäytetyön kontekstiin liittyviä työtehtäviä

Torstai 6.10.2022

Tavoite: Suunnitella miten POC-hankkeen kanssa jatketaan. Tänään on myös säännöllinen POC-jatkopalaveri asiakkaan kanssa. Mahdollisesti testata Jirassa olevien virheellisten tehtävien toiminta uudestaan nyt kun teknistä toteutusta on viilattu.

Toteutuminen: Pidimme asiakkaan kanssa jatkopalaverin ja sovimme etenemisestä. Kuten olen jo aiemmin pohtinut, minusta olisi hyvä, että POC-hankkeella olisi jokin vetäjä ja tätä mieltä oli myös toimialan business partner. Hän sanoi olevansa yhteydessä toimialan palvelujohtajaan ja kysyvänsä, että kuka voisi sopia asiakkaan puolelta vetämään hanketta, jottei se olisi niin isosti minun ja ratkaisuarkkitehdin varassa. Pohdimme palaverissa myös suorituskykyasioita, kuten stressitestauksia, jotta nähdään, että sovellus toimii myös isommilla volyyymeillä.

Kävin palaverin jälkeen tietokantaspecialistin kanssa asiakkaan ERP-järjestelmän Oracle-tietokannan suorituskyvystä keskustelua. Minun huoleni oli se, ettei Oraclen tietokanta pystyisikään vastaamaan sovelluksesta tuleviin kutsuihin riittävässä ajassa vaan siitä muodostuisi asiakkaan ERP-järjestelmän tietokantasuorituksiin jonoa. Tämä on sellaista aluetta Oraclen tietokannoista ja tietokannoista ylipäätään, josta olen aiemmin päässyt vain raapaisemaan pintaa. Onneksi on tietokantaspecialistin kaltaisia asiantuntijoita, joiden kanssa asiasta voi sparrata.

Keskustelun alussa tietokantaspecialisti hieman säikäytti minut sanomalla, että siitä voi tulla ongelma ja kun kysyin, että mitä tälle voi tehdä, että kaatuuko hanke tähän kokonaan, niin siihenkin vastaus oli, että voi kaatua. En kuitenkaan hyväksynyt vastausta sellaisenaan vaan jatkoin asian puimista ja hieman myöhemmin siirryimme testaamaan asiaa käytännössä.

Testasimme ensiksi Azuren samanaikaisten ajojen suoritusta ja totesimme, että kaikki kutsut, jotka osuvat integraation aikaleiman sisään samalle sekunnille suorittavat saman aikaisesti. Voi siis olla esimerkiksi tilanne, että reilusta tuhannesta käyttäjästä vaikka 30 käyttäjää tekee ensimmäisen työtilauksen leimauksen 6.00.01 ja seuraavat 50 käyttäjää 6.00.03 jolloin molemmat suorittavat omissa erissään riittävän nopeasti eikä ongelmia pitäisi syntyä. Tämän lisäksi tarkistimme myös Oraclen tietokannasta, ettei sinne jää inaktiivisia sessioita integraatiokäyttäjän tekemistä kutsuista. Tulimme myös tässä testissä siihen tulokseen, ettei jää eli kaikki pitäisi sen osalta myös olla kunnossa.

Lopulta erilaisten testauksien jälkeen tulimme yhdessä tietokantaspecialistin ja mukaan liittyneen integraatiokehittäjän kanssa tämän päivän testauksien osalta siihen lopputulokseen, että sovelluksen pitäisi pystyä käsittelemään pyynnöt riittävässä ajassa tuotantokäyttöä varten. Sovimme, että

teemme vielä lähiaikoina systemaattisempaa testausta ja dokumentoimme testauksen, jossa tehdään Power Appsista muutoksia ja katsotaan, että ne reflektoituvat ERP-järjestelmän puolelle riittävässä ajassa.

Perjantai 7.10.2022

Tavoite: Aiemmin havaitun kohdeintegraation hitauden korjaaminen.

Toteutuminen: Opinnäytetyön kontekstiin kuuluvista työtehtävistä suurimman osan vei kohde IAL-näkymän hitauden tutkiminen ja kyseisen IAL-näkymän optimoiminen. Näkymään haetaan eri tauluista tietoja, jotka muodostavat kokonaisuuden, josta löytyy asiakkaan kohdetiedot. Kyseistä IAL-näkymää hyödynnetään monissa integraatioissa tuotannossa ja päätimme, että hyödynnämme samaa IAL:ää myös *customer_object*-tauluun, johon tuodaan vastaavaa dataa, kun vaikkapa tuotannon kohdeintegraatioissa.

IAL on luotu asiakkaan ERP-järjestelmän alkuaikoina ja siinä on ollut ongelmia pidemmän aikaa. Keväällä siihen tehtiin korjaus, jonka oli tavoitteena nopeuttaa sitä ja tuoda lisää näkyvyyttä prosessin mukaiseen kohteiden ja sopimusten käsittelyyn. Parannus ei kuitenkaan toiminut toivotulla tavalla vaan se aiheutti integraation hitausongelman, jonka vuoksi integraation tekemät haut alkoivat yhtäkkiä kestämään tunteja. Tavoitteena parannuksella oli mahdollistaa kohteiden säännöllinen päivittyminen taustajärjestelmiin viemällä ainoastaan muuttuneiden kohteiden tiedot joko kohde- tai sopimustietoihin tehdyn muutoksen jälkeen.

Poistin keväällä tehdyn parannuksen ja rupesin pohtimaan kuinka IAL:n saisi mahdollisimman optimaaliseksi mutta silti tuomaan samat tiedot kuin aiemmin. Tein useita erilaisia versioita ja kokeilin vaihtoehtoja laidasta laitaan. Monien erilaisten variaatioiden kokeilun jälkeen löysin hitauden syyn ja syyksi lopulta osoittautui ERP-järjestelmään muokkauksena tehty metodikutsu, jonka avulla haettiin sopimuksen integraatiotäppää sekä sopimuksen olemassaolon tarkistusta.

Ongelmat eivät kuitenkaan siihen loppuneet sillä IAL object developer -työkalu ei huolinutkaan uudelleen muotoilemani SQL-kyselyä vaan antoi virheen IAL-taulun muodostamisesta kertomatta ORA-virheen lisäksi tarkemmin mistä on kyse. Ensiksi epäilin uudelleen muotoilemani kyselyn olevan liian pitkä, jos IAL:ssä olisi esimerkiksi 4000 merkin maksimimäärä. Tällaisia on aiemminkin tullut vastaan asiakkaan ERP-järjestelmän kanssa, joten epäilykseni olisi voinut osua oikeaan. Näin ei kuitenkaan ollut, sillä IAL object developer nielaisi toisen vielä pidemmän kyselyn ilman virheilmoitusta. Sen jälkeen rupesin pohtimaan tekemiäni muutoksia ja kävin ne järjestelmällisesti ja yksitellen läpi. Lopulta paljastui, että yhden kentän poistamalla SQL:n valittavista sarakkeista, sain

koko muun kyselyn suoraan tallennettua. Kyseisen kentän olin hakenut alkuperäistä kyselyä eroavalla tavalla, joten sen vuoksi se oli aiemmin toiminut. Vaihdoin siis kyseisen kentän käyttämään alkuperäistä metodia, jonka jälkeen IAL:n pystyi julkaisemaan ongelmitta.

Viikkoanalyysi 6

Viime viikon onnistuneen liiketoimintademon ja sen jälkeisen positiivisen palautteen jälkeen keskityimme taas enemmän POC-hankkeen seuraaviin vaiheisiin. Kävimme heti alkuvuikosta business partnerin ja sovellusarkkitehdin kanssa läpi demon jälkeisiä tunnelmia. Liiketoiminnat olivat selvästi kiinnostuneita ja mainitsemamme mahdollisuudet Power Appsin hyödyntämisestä muuhun olivat myös kiirineet muiden liiketoimintojen tietoon. Saman toimialan, toisen liiketoiminnan edustajat olivat jo kyselleet business partnerilta, että milloin heille järjestetään demo asiasta. Esityksessä asiasta oli vain pieni maininta Power Appsin mahdollisuuksista puhuttaessa. Asia oli kuitenkin lähtenyt leviämään ja osittain sen leviämistä oli vauhdittanut yhden liiketoimintademossa olleen viestintähenkilön innokkuus sovelluksen mainostamisesta Yammer-viestikanavassa. Mainostaminen tuli business partnerillekin puskista eikä hän ollut näin avoimen mainostamisen kannoilla vielä, jos sovelluksen kanssa tulisivat jotain teknisiä tai taloudellisia haasteita vastaan. Siinä tapauksessa siitä voisi aiheutua pettymystä, kun luvattua parannusta ei saataisikaan. Toisaalta olisi kuitenkin hyvä kommunikoida loppukäyttäjien kanssa ja saada heidät ajoissa mukaan (Dell 2019). Loppukäyttäjien kanssa kommunikoidessa on tärkeä pitää teksti mahdollisimman simppeleinä ja mennä suoraan asiaan. Viestinnässä kannattaa mielellään hyödyntää yrityksen sisäisiä viestintäkanavia sähköpostin sijasta kuten viestintähenkilö oli tehnytkin ja testata viestin ymmärrettävyys ennen lähettämistä. (Ihana 2017.)

Keskustelussa yhtenä tärkeimmistä asioista jatkon osalta nousi esille isomman volyymin testauksen tärkeys. Nythän olemme pystyneet demoamaan, että tiedot liikkuvat järjestelmien välillä yksittäisten esimerkkien ja tapausten osalta. Totesimme kaikki yhteen ääneen, että seuraavaksi on todella tärkeä pystyä varmistamaan, että suorituskyky pysyy mukana myös kuorman kasvaessa. Totesimme myös, että pyrimme mahdollisimman pian siirtymään vaiheeseen, jossa asiakkaan työntekijät pääsevät osallistumaan testaamiseen. Suurin osa perustoiminnallisuuksista on jo kuitenkin olemassa, joten olisi tärkeää saada palautetta kentän työntekijöiltä sen sijaan että hioisimme ja kehittäisimme sovellusta liian pitkälle pelkän ICT:n voimin. Sovelluksen tulisi kuitenkin lähtökohtaisesti tuottaa liiketoiminnoille lisäarvoa ja parhaimman palautteen saa käyttäjiltä, joiden olisi tarkoitus käyttää sovellusta päivittäisten työtehtävien hoitamiseen. (Vossen 2018.)

Power Appsin osalta kuormantestaamisella viitataan lähinnä taustajärjestelmien suorituskykyyn, sillä Power Appsissa jokainen käyttäjä on omassa instanssissaan eli käyttäjien määrällä ei ole vaikutusta Power Apps sovelluksen suorituskykyyn. Vaikutus tulee esille käyttäjien tekemistä toiminnoista kuten tietolähteisiin tehtävistä kutsuista. (Li 2019.) Kävin kuormantestaukseen liittyen keskustelua tietokantaspecialistin kanssa ja pohdimme voisiko suorituskyvyssä tulla haasteita vastaan sitten kun käyttäjiä ja käyttäjien tekemiä kutsuja olisi saman verran, kuin tuotannossa nyt. Tarkistettavina asioina oli Azure Logic Appsin samanaikaisten suoritusten kestot sekä Oraclen suorituskyky isommilla määrillä. Tässä kappaleessa käytin termiä kuormantestaus, joka kuvaa yhtä tyyppiä erilaisten suorituskykytestien joukossa. Kuormantestauksen kanssa usein sekaisin käytettävä stressitestaus on myös yksi suorituskykytestauksen tyypeistä ja on hyvä ymmärtää näiden kahden erot. Siinä missä kuormantestauksella testataan sovelluksen tai nettisivun kuormitusta suurella mutta realistisella määrällä käyttäjiä, stressitestauksessa mennään pidemmälle ja pyritään löytämään yläraja. (Cohen 2022.)

Toisen liiketoiminnan sovelluksen uusimisen mahdollisuuden Power Appsiin heräsi myös yrityksen toiminnanjohtaja, joka rupesi pohtimaan, että olisiko mahdollista saada isompi osuus kehityksestä meille. Hänellä vaikutti olevan myös pelko, että asiakkaan kumppaniyritys yrittäisi vastaavasti saada kehityksen kokonaan heille. Ymmärrän kilpailuaseteleman mutta tällä hetkellä Power Apps kehitys vaikuttaisi olevan automaattisesti tulossa minulle eli meille. Liiketoiminta mielessä en liikaa haluaisi lähteä sotkemaan kuvioita liiallisella myymisellä tai asiaan puuttumisella. En myöskään haluaisi sotkea hyvää yhteistyötä asiakkaan kumppaniyrityksen kanssa kummankaan puolen ahnehtimisen takia. Sen sijaan, että kilpailisimme molemmat saman asian kokonaisuudesta haltuunotosta, voisimme jatkaa hyvää kilpailullista yhteistyötä, kasvattaa molempien osaamista uusien teknologioiden parissa sekä sen kautta molemmat hyötyä projektista. (Hamel, Doz & Prahalad 1989.)

Olen POC-hankkeen kehityksen aikana pohtinut useaan otteeseen, että onko Power Apps varmasti asiakkaalle hyvä valinta nykyisen mobiilisovelluksen tilalle. Nyt kun asiakas on erittäin kiinnostunut jatkamaan hankkeen kanssa, niin tämä ajatus konkretisoituu entisestään. Olen päiväkirjatyön aikana lueskellut monenlaisia juttuja aiheeseen liittyen ja löytänyt sekä puolesta että vastaan argumentteja. Vasta-argumentit liittyvät muun muassa sovelluksen suorituskykyyn, sovelluksen jatkamiseen, Microsoftin tekemiin päivityksiin, jotka saattavat rikkoa jotakin sekä kustannuksiin (Skyline consultants s.a.). Lisäksi sovelluksen kompleksisuutta, tyyliä ja versionhallintaa tulee pohtia (Fisette 2021).

Oma ajatuksenjuoksuni lähtee kuitenkin siitä, että minut on edellisen business partnerin toimesta valittu kehittämään sovellusta ja pääfokus omassa tekemisessäni on ollut siinä, että saan toteutettua tiimin kanssa yhteistyössä teknisen toteutuksen, jonka avulla voisimme arvioida Power Appsin

soveltuvuutta nykyisen sovelluksen korvaajana. Asiakas on ennen hankkeen käynnistämistä myös tehnyt asiakkaan kumppaniyrityksen johdolla strategiatyön ja kartoittanut eri vaihtoehtoja nykyisten mobiiliratkaisujen parantamiseen. Olen siis alusta asti ajatellut, että asiakas on tehnyt teknisen due diligencen huolella eikä minun tarvitse siihen puoleen paneutua kovin syvällisesti. (Kubrak 2021.) Se mikä vahvasti omassa mielessäni puoltaa Power Appsia, on ulkopuolisen Power Apps konsultti-firman erittäin kokeneen Microsoft MVP-tason neuvonantajan mukanaolo ilman soraääntä. Hän ei ole missään vaiheessa keskusteluissamme ilmaissut, ettei Power Apps soveltuisi asiakkaan liiketoimintakriittisiin käyttötapauksiin. Hän on aina pikemminkin ilmaissut, että kyllä asiat onnistuvat Power Appsilla ja pystynyt niitä myös todistamaan dokumenttien, materiaalien ja muun faktatiedon avulla.

Seuraavalla viikolla minun täytyy ehdottomasti keskittyä myös HR häirintäilmoitukseen ja ruveta siirtämään sitä tuotantoon, jotta ehdimme testaamaan ja varmistamaan, että se varmasti toimii tuotannossa ennen HR-henkilöstölle pidettävää demotilaisuutta. Sain tällä viikolla HRIS-spesialistilta vihdoinkin vihreää valoa tuotantoon siirron osalta, vaikkakin edelleen HR:n kanta oli, että yleisten järjestelmänvalvojen oikeuksia pitäisi pystyä rajaamaan. Heidän mielestään tämä tapahtuisi poistamalla yleisiltä järjestelmänvalvoilta oikeudet kyseiseen ympäristöön sekä varmistamalla auditoinneilla, ettei yleiset järjestelmänvalvojat laita oikeuksia takaisin tai muuten tarkastele kyseiseen sovellukseen liittyen mitään tietoja. Auditointi voisi onnistua mutta oikeuksien rajaamisen osalta en osaa sanoa. Aion käydä ensi viikolla asiakkaan ICT:n palvelun toimituksesta vastaavan kanssa asiaa läpi, sillä hänellä on yleiset järjestelmänvalvojan oikeudet, ja sitä kautta pääsen tarkastelemaan mitä hän näkee kyseiseen sovellukseen liittyen.

3.7 Seurantaviikko 7 (vko 41)

Viikon tavoitteena on päästä HR Häirintäilmoituksen kanssa eteenpäin ymmärtämällä mitä kaikkea sovelluksen siirtämiseen kehitysympäristöstä tuotantoympäristöön vaaditaan. Tähän kuuluu muun muassa ALM-prosessin ymmärryksen kasvattaminen Power Apps kontekstissa sekä tarvittavien oikeuksien hankkiminen uudelle HR Häirintäilmoitus palvelutilille, sillä tämä jäi aiemmin osittain kesken. Lisäksi tavoitteena on saada valmiiksi POC-hankkeen tehtävälistalla olevat keskeneräiset asiat. Emme ole vielä saaneet asiakkaalta vahvistusta, miten ja kenen johdolla POC-hanketta aiotaan jatkaa, joten pyrimme testaamaan, hiomaan ja korjaamaan nykyistä toteutusta ennen uusien kehitysten aloittamista. Viime viikolla korjattua kohdeintegraatiota on myös tavoite testata tällä viikolla, jotta korjaus saadaan mahdollisimman pian tuotantoon. Kyseisellä korjauksella on moniulotteinen vaikutus, sillä IAL-näkymää käytetään monessa paikassa, muun muassa myös POC-hankkeessa.

Maanantai 10.10.2022

Tavoite: Viimeistellä asiakas- ja kohdeintegraatio, joka on yksi viimeisistä auki olevista tehtävistä Jirassa uudelleen testausten lisäksi.

Toteutuminen: Saimme kuulla asiakkaan kanssa pidettävässä ICT tilanne check -palaverissa, että liiketoiminnoilla on työn alla ja etsinnässä tuoteomistaja POC-hankkeelle. Business partner sanoi, että kehitysjohtaja oli luvannut tämän viikon keskiviikkona ilmoittaa, kuka tulee ottamaan POC-hankkeen jatkon vastuulle heidän puoleltaan.

Pidimme tilanne check-palaverin jälkeen lyhyen päivittäispalaverin, jossa kävimme läpi tämänhetkisen tilanteen. Sovimme, että keskeytämme päivittäispalaverit hetkeksi, sillä nyt odotellessamme POC-hankkeen jatkon suuntaviivoja, ei ole kovinkaan paljon läpikäytävää eikä sen vuoksi myöskään tarvetta pitää päivittäispalaveria.

Integraatioiden osalta meillä oli vielä viimeistelemättä asiakas- ja kohdeintegraatio. Sovimme palaverissa, että viimeistelemme sen muun muassa poistamalla integraatiosta tarpeettomia toimintoja ja päivittämällä integraation kyselylauseke tuomaan enemmän testidataa ERP-järjestelmästä Data-verseen.

Tiistai 11.10.2022

Tavoite: Aloittaa HR Häirintäilmoituksen tuotantoon siirto muun muassa siirtämällä ratkaisupaketti kehitysympäristöstä tuotantoympäristöön.

Toteutuminen: Huomasin heti siirtoa aloittaessani, ettei siirto tule suorilta onnistumaan, sillä uudelta palvelutililtä näytti puuttuvan oikeudet kaikkiin ympäristöihin. Minulla ei myöskään ollut mitään muuta käyttäjää, jolla olisin päässyt itse antamaan oikean ympäristön käyttöoikeudet.

Ennen integraatiopään muutosta, testailimme asiakkaan ERP-järjestelmässä uusien ja muuttuneiden kohteiden näkymistä IAL-näkymässä (kuva 9). Totesimme kaikkien tekemiemme testitapausten muutosten heijastuvan myös IAL-näkymään, joten tulimme siihen tulokseen, että muutos on onnistunut ja se voidaan viedä eteenpäin. Olimme pohtineet, että rajaamme IAL-näkymään haettavia kohteita sopimuksen tiedoissa olevan integraatiotäpän avulla. Eli ainoastaan ne kohteet, joiden sopimuksella on integraatiotäppä näytetään integraatiossa. Tällä sekä vähennämme IAL-näkymään tulevien kohteiden määrää, vähennämme tuotannossa turhia virheilmoituksia, että selkeytämme integraation logiikkaa kaikkiin suuntiin.

The screenshot shows an Oracle SQL Developer window with the following SQL query:

```

SELECT
EO_MCH_CODE, EO_MCH_NAME, EO_SUP_MCH_CODE, EO_OBJ_LEVEL, EOA_ADDRESS1, EOA_ADDRESS3, EOA_ADDRESS4, EOA_ADDRESS7,
CI_ASSOCIATION_NO, EO_CONTRACT, EOP_NAME, EOP_IDENTITY, CONTRACT_EXISTS, INTEGRATION_SERVICE,
CUSTOMER_ERP_INTEGRATED, C_CUST_DOC_CLASS_EXIST, C_CUSTOMER_ERP, EO_OPERATIONAL_STATUS,
TRANSFER_TO_WEB, TRANSFER_TO_FATMAN, C_PREV_CUST_OBJ, C_EXPIRY_DATE

FROM ifsinfo.c_fimx_objects

WHERE
(
EO_C_TIMESTAMP > trunc(sysdate, 'dd') OR
EOA_C_TIMESTAMP > trunc(sysdate, 'dd') OR
EOP_C_TIMESTAMP > trunc(sysdate, 'dd') OR
contract_header_timestamp > trunc(sysdate, 'dd') OR
contract_row_timestamp > trunc(sysdate, 'dd') OR
contract_org_timestamp > trunc(sysdate, 'dd') OR
contract_object_timestamp > trunc(sysdate, 'dd')
)

```

The results table displays 16 rows of data. The columns are: EO_MCH_CODE, EO_MCH_NAME, EO_SUP_MCH_CODE, EO_OBJ_LEVEL, EOA_ADDRESS1, EOA_ADDRESS3, EOA_ADDRESS4, EOA_ADDRESS7, and CI_ASSOCIATION_NO. The data includes contract codes, names, levels (Asiakaskohde, Työkohde), addresses, and association numbers.

EO_MCH_CODE	EO_MCH_NAME	EO_SUP_MCH_CODE	EO_OBJ_LEVEL	EOA_ADDRESS1	EOA_ADDRESS3	EOA_ADDRESS4	EOA_ADDRESS7	CI_ASSOCIATION_NO
1	999000122	...	Asiakaskohde	...	40100	JYVÄSKYLÄ	FI	...
2	999000277	...	Asiakaskohde	...	90830	HAUKIPUDAS	FI	...
3	999000307	...	Asiakaskohde	...	90830	HAUKIPUDAS	FI	...
4	998394841	...	Asiakaskohde	...	02100	ESPOO	FI	...
5	999435453	...	Asiakaskohde	...	91800	TYRNÄVÄ	FI	...
6	998419428	...	Asiakaskohde	...	70200	KUOPIO	FI	...
7	999000083	...	Asiakaskohde	...	40320	JYVÄSKYLÄ	FI	...
8	100016915	998394841	Työkohde	...	02100	ESPOO	FI	...
9	100024161	999435453	Työkohde	...	91800	TYRNÄVÄ	FI	...
10	100106033	999000083	Työkohde	...	40320	JYVÄSKYLÄ	FI	...
11	10032862	999000122	Työkohde	...	40100	JYVÄSKYLÄ	FI	...
12	10036886	999000122	Työkohde	...	40100	JYVÄSKYLÄ	FI	...
13	10036887	999000122	Työkohde	...	40100	JYVÄSKYLÄ	FI	...
14	10062484	999000277	Työkohde	...	90830	HAUKIPUDAS	FI	...
15	10062506	999000307	Työkohde	...	90830	HAUKIPUDAS	FI	...
16	10062963	999000307	Työkohde	...	90830	HAUKIPUDAS	FI	...

The status bar at the bottom indicates: 16 rows selected in 26,681 seconds.

Kuva 9. IAL-näkymän SQL-kysely ja tulokset PL/SQL-developer työkalussa (Oracle PL/SQL developer 2022)

Tässä samassa yhteydessä löysimme asiakkaalta paljon sopimuksia, joilla ei ollut integraatiotäppää lainkaan. Löydös oli hyvä, sillä pahimmillaan integraatiotäppän puuttuminen voi aiheuttaa työtilauksen virhetilanteen, jolloin jokin tärkeä työ saattaa jäädä tekemättä sen vuoksi, ettei integraatio siirrä työtilausta järjestelmien välillä. Ohjeistimme myös integraatiokehittäjää tekemään muutoksen kohdeintegraation SSIS-puolelle, jotta testin suorittaminen kokonaisuudessaan voidaan toteuttaa.

Keskiviikko 12.10.2022

Tavoite: Jatkaa HR Häirintäilmoituksen tuotantoon vientiä.

Toteutuminen: Tutustuin tarkemmin Power Appsin tapaan käsitellä kehitys- ja tuotantoympäristön versioita. Olimme aiemmin käyneet ulkoisen Power Apps konsultin kanssa pääpiirteittäin pikaisessa palaverissa läpi, miten siirron tulisi tapahtua, mutta nyt siirtoa aloittaessa minulla heräsi kysymyksiä siirron toteuttamistavoista. Googlailin ja katselin videoita aiheeseen liittyen ja sitä kautta kasvatin ymmärrystäni siirtoon liittyvästä prosessista. Pohdin tuotantoympäristöön käytettävää siirrotapaa hallitun ja ei-hallitun välillä. Hallittuna siirtäminen kävisi paremmin yhteen ALM-prosessin kanssa mutten ollut varma, miten kätevimmin saan siirrettyä kaikki viittaukset Dataverse-tauluihin,

Power Automaten työnkulkuihin sekä Microsoft Formsiin liittyen omalta tunnukseltani uudelle palvelutilille (service account). Ei-hallittuna pystyisin paremmin kontrolloimaan ratkaisua myös tuotannon puolella.

Torstai 13.10.2022

Tavoite: Uudelleen testata POC-hankkeen epäonnistuneita testitapauksia sekä saada valmiiksi kaikki Jiran keskeneräiset työt, jotta pääsemme suoraan jatkamaan uusilla asioilla, kun asiakas on saanut selvyuden tulevasta tuoteomistajasta.

Toteutuminen: Testasimme uudelleen työtilauksen käsittelyn epäonnistuneita tapauksia. Saimme epäonnistuneet tapaukset kulkemaan joko suoraan tai pienellä viilauksella. Tein kaikki tarvittavat muutokset Logic Appsiin ja olin tyytyväinen, kun sain ne melko vaivatta tehtyä. Se toi varmuutta ja luottoa Logic Appsin kehittämiseen jatkossakin.

Illtapäivällä juuri ennen töiden lopetusta huomasin, että minua on pyydetty osallistumaan tuotannossa aamupäivällä alkaneen laajavaikutteisen häiriön selvitykseen DS:ään liittyen. Sen selvitykseen menikin sitten loppuilta.

Perjantai 14.10.2022

Asiakkaan tuotannossa eilen havaitun laajavaikutteisen häiriön uusiutuminen ja sen selvittely vei ison osan päivästä. Tutkin tämän lisäksi lisää ALM-prosessista mutten ehtinyt / pystynyt tekemään mitään konkreettisia toimenpiteitä oikeuksien puuttuessa.

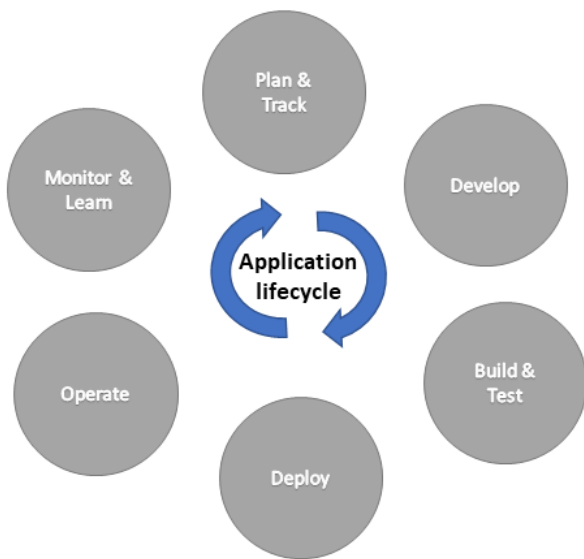
Viikkoanalyysi 7

Viikko oli kiireinen asiakkaan tuotantohaasteiden ja kehitysten kanssa. Tavoitteena oli päästä aloittamaan HR Häirintäilmoituksen tuotantoon siirto sekä tyhjentää POC-hankkeen lista kehityksen seuraavaa vaihetta odotellessa. Jälkimmäinen onnistui mainiosti testattuamme aiemmin epäonnistuneet työtilauksen käsittelyn testitapaukset uudelleen ja todettuamme niiden toimivan riittäväällä tasolla nykytilanteeseen nähden. Jotkin tapauksista toimivat suoraan ensimmäisellä testauksella, jotkin tapaukset vaativat pieniä muutoksia Logic Appsin työnkulkuihin, jotta prosessi saatiin kulkemaan oikealla tavalla. Samassa yhteydessä myös hienosäädin muita pienempiä havaittuja epäkoh-tia.

Toisen tavoitteen osalta oli haasteita, sillä HR Häirintäilmoituksen siirtoa kehitysympäristöstä tuotantoympäristöön hankaloittivat tuotannon kriittiset ongelmat, siirron aloituksessa havaitsemani käyttöoikeushaasteet ja siirtoon tarvittavan ymmärryksen aukot tietämyksessäni. Käyttöoikeus- ja ymmärryshaasteet ilmenivät heti kun aloitin tuotantoon siirtoa. Ensimmäinen haaste oli ympäristön

avaaminen. Uudelta HR Häirintäilmoitus käyttäjältä puuttui ympäristöjen oikeudet kokonaan eikä pystynyt niitä antamaan. Sain käyttöoikeudet hankittua hieman myöhemmin, mutta tuotannon laajavaikuttaiset häiriöt veivät kahdesta päivästä ison osan ajasta ongelmien selvittelyyn, joten HR Häirintäilmoitukselle jäi selvästi vähemmän aikaa. Huomasin myös olevani prosessista hieman epävarma ja jouduin pohtimaan sekä selvittämään, että miten siirto kannattaisi järkevimmin tehdä. Pohdin muun muassa, että pitäisikö sovellus siirtää hallittuna (managed) vai ei-hallittuna (unmanaged), kuinka saisin ratkaisun komponenttien omistajuuden siirrettyä omalta sekä Power Apps palvelutililtä uudelle HR Häirintäilmoitus palvelutilille. Näitä komponentteja ovat muun muassa Microsoft Formsilla luotu lomake, Power Automatella luodut työnkulut sekä Power Appsin sisällä olevat Dataversen taulut, valinnat (choice), käyttöoikeusryhmät ja tietenkin Canvas-sovellus.

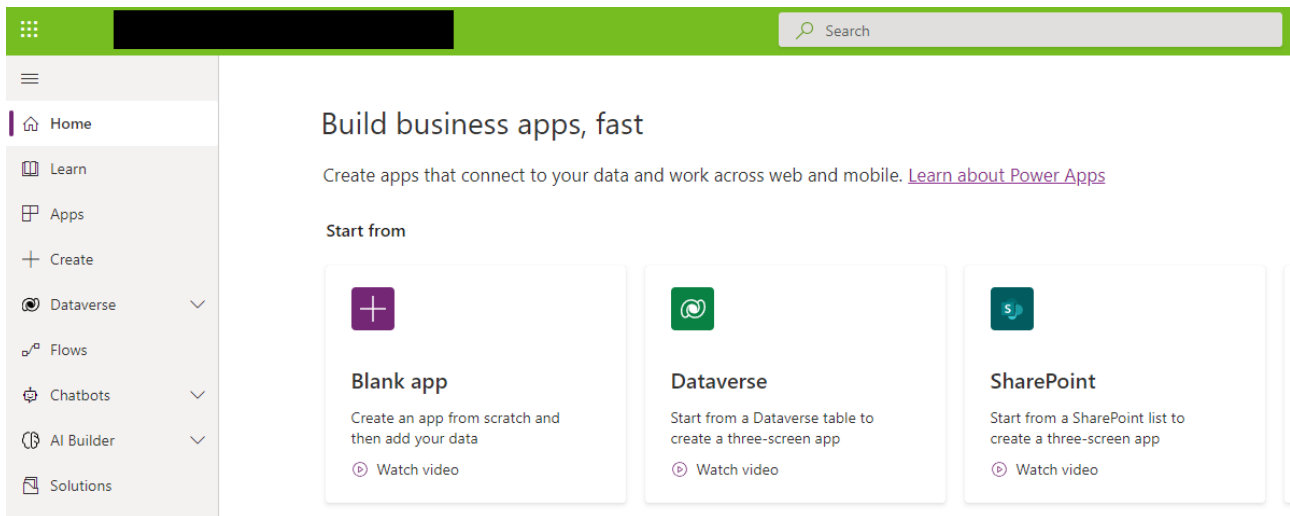
Ympäristöjen väliseen siirtoon liittyy käsite ALM ja yritin sitä avata itselleni tutkimalla siihen liittyviä kirjoituksia ja videoita. ALM kattaa sovelluksen elinkaaren, johon kuuluu hallinnointi, kehitys ja ylläpito (Microsoft 2022f).



Kuva 10. Sovelluksen elinkaari (Microsoft 2022f.)

ALM:n tärkein tavoite on luoda tehokkuutta ennakoitavaan ja toistettavaan sovelluskehitykseen (Microsoft 2022f.) Jotta voi ymmärtää elinkaaren hallintaa Power Appsin kontekstissa, on hyvä yrittää ymmärtää mistä ja miten Power Apps rakentuu. Kaikki mitä Power Appsisä tai Power Automatella rakennetaan sisältyvät ympäristöön. Sekä Microsoftin että esimerkiksi MVP Reza Dorraniin suosittelema paras käytäntö sovellusten ja kokonaisuuksien rakentamiselle on ratkaisujen hyödyntäminen. (Dorrani 2022.) Tämä on täysin epäintuitiivinen ja ilman tutustumista tai ohjeistusta hankala käytäntö omaksua, varsinkin aloittelevana Power Apps tekijänä. Aloitin ensimmäisen kerran

Power Appsiin tutustumisen vajaa vuosi sitten ja ensimmäinen asia, jonka tein Power Appsiin kirjaututtua oli ratkaisuun sisällyttämisen sovelluksen luominen suoraan aloitusnäytöltä.



Kuva 11. Aloitusivu kirjautumisen jälkeen (Microsoft Power Apps 2022b)

Power Appsin aloitusivun valinnat houkuttelevat valitsemaan joko tyhjän sovelluksen tai Dataversen tietokantataulun luomisen ensiksi, vaikka hampurilaisvalikon ”Solutions” olisi suositeltu tapa edetä. Onneksi silloinen business partner osasi heti ohjeistaa ratkaisun hyödyntämisestä ennen kuin ehdin pidemmälle pelkän ratkaisusta irrallaan olevan toteutuksen kanssa. Ei kuitenkaan ole maailmanloppu, jos sovelluksen tekee ratkaisun ulkopuolelle, sillä sen voi helposti siirtää ratkaisun sisälle. On vain selkeämpää aloittaa ratkaisun kanssa, jolloin kaikki tulee valmiiksi ratkaisun sisälle eikä tarvitse tehdä ylimääräisiä toimenpiteitä. (Dorrani 2022.)

Olin ajatellut, että HR Häirintäilmoituksen ratkaisun omistajaksi jää Power Platform palvelutili ja sitä käytettäisiin myös tuotannossa. Silloin nykyisiin komponentteihin ei olisi tarvittu juurikaan muutoksia. Nyt kuitenkin muutos piti tehdä eikä se hieman odotusten vastaisesti onnistunutkaan kaikkialle yhdellä kertaa, vaan vaati hieman enemmän työtä. Löysin YouTubesta monenlaisia videoita, joissa ohjeistettiin omistajuuden siirtämisestä käyttäjältä toiselle Power Automaten, Cmdlet-komentojen ja PowerShellin avulla. Metodien pitäisi toimia jopa sellaisissa tilanteissa, joissa omistaja on lähtenyt organisaatiosta. Niissä tapauksissa ajon suorittajalla tulee kuitenkin olla yleiset järjestelmänvalvojan tai Power Platform järjestelmänvalvojan -tasoiset oikeudet. (Devaney 2022.) Tässä tapauksessa kyse oli kahdesta tilistä, jotka ovat edelleen aktiivisia, joten pääsin tekemään muutoksen uudella HR palvelutilillä suoraan. Alla olevassa kuvassa näkyy Power Automate työnkulku ja siihen liittyvät vaiheet, jolla muutos saadaan aikaiseksi. Työnkulun helpottamiseksi ID:n syöttämisen sijasta on käytetty Office 365 liitintä (connector), jonka avulla ID on haettu käyttäjän sähköpostia hyödyntäen. (Dunnam 2022.)

The image shows a screenshot of a Microsoft Power Automate flow with three steps:

- Manually trigger a flow**: The first step, highlighted with a blue border. It includes a hand icon and a blue water drop icon.
- Get user profile (V2)**: The second step, highlighted with an orange border. It has a user profile icon and a search icon. The input field for "User (UPN)" contains the text "svc_hrhairintailmoitus@" followed by a redacted area. Below the input field is a "Show advanced options" dropdown menu.
- Set App Owner**: The third step, highlighted with a purple border. It has a gear icon and a search icon. The input fields contain the following values:
 - * Environment Name: 8c1bd185-dba6-4af6-8ea7-67da3a75c6d9
 - * PowerApp Name: 78062bfd-a855-4747-9b20-0a4d037d37c5
 - API Version: 2016-11-01
 - Content Type: application/json
 - Role For Old App Owner: CanView
 - New PowerApp Owner: A user card with a red square icon, the text "Id", and a close button "x".

Kuva 12. Power Apps sovelluksen omistajan muutos Power Automatella (Microsoft Power Automate 2022c)

Huomasin käytännön siirtämistestailujen kautta, että lisää tarvittavia oikeuksia ja lisenssejä puuttui. Käyttäjältä muun muassa puuttui sekä Power Appsin että Power Automaten "per app user plan" -lisenssit ja tämän lisäksi Microsoft Formsin sisältävä Office-lisenssi. Olin esimerkiksi siirtämässä "Notification Deletion" työnkulkua HR:n palvelutilille, mutta siirto epäonnistui puuttuvien oikeuksien takia ja Power Automate antoi alla olevan virheen:

Details ✕

⊗ Flow client error returned with status code "Forbidden" and details [{"error": {"code": "MissingAdequateQuotaPolicy", "message": "Käyttäjällä b4053e68-975b-48f2-8159-fe8fdb7bccff ei ole palvelupakettia, joka sopii muulle kuin vakioyhteydelle Microsoft Dataverse. https://go.microsoft.com/fwlink/?linkid=2123710", "extendedData": {}}}]

Flow name

Notification Deletion

Description

Provide a helpful description explaining how your flow works

Owner *

Only owner with a premium license will be able to run this flow. [Learn more](#)

H # HR hairintailmoitus service ac... ✕

The owner of this flow needs a premium license. Contact your admin to assign a license or ask the new owner to start a free trial or [self-service](#) purchase a Power Automate plan.

Plan *

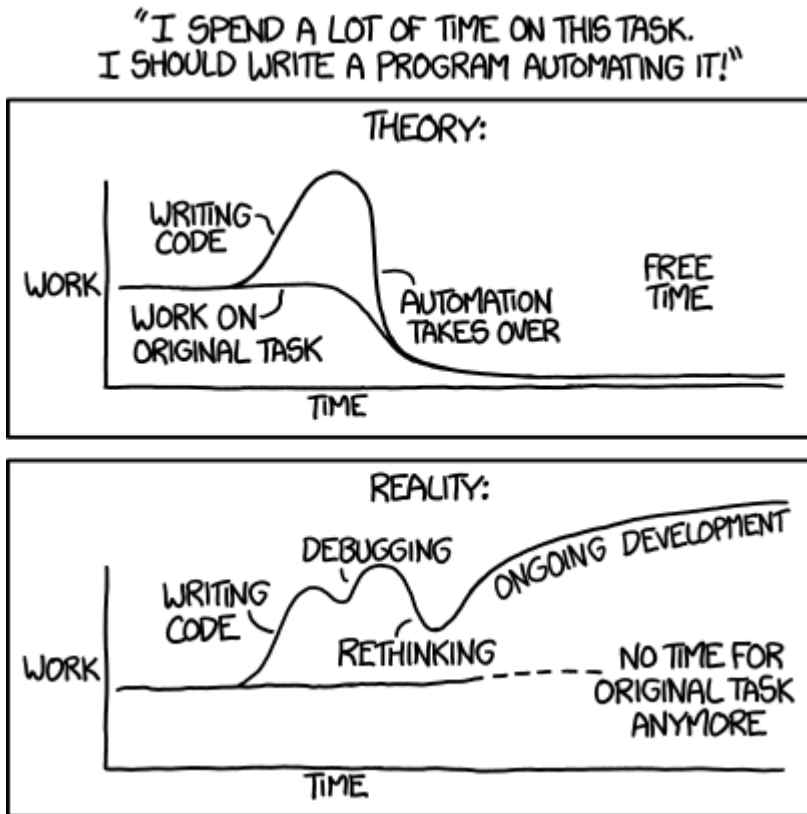
Owner

Per-flow (There's no capacity allocated for this environment.)

Save **Cancel**

Kuva 13. Virhe Power Automaten omistajuuden muutoksessa (Microsoft Power Automate 2022d)

Tutkiessani lisää sovelluksen elinkaaresta Power Apps kontekstissa, törmäsin blogikirjoitukseen, jossa pohdittiin, onko Power Apps vielä valmis ALM:ään. Kirjoituksen sanomaa pohtiessa ja peila-
tessa HR Häirintäilmoitus sovellukseen sekä sen tavoitteisiin, kävi selväksi, ettei mikään raskas
ALM-prosessi sovi HR Häirintäilmoituksen kanssa. ALM voi toimiessaan tuoda säästöä ajankäytön
osalta mutta ALM:n hyödyntäminen voi pienemmässä mittakaavassa olla liian työlästä ja hyödyt
voi jäädä saavuttamatta. (Niiranen 2020.)



Kuva 14. Automaatio vertaus (xkcd.com s.a.)

Niiranen (2020) oli käyttänyt xkcd:n tekemää piirroskuvaa havainnollistamaan mielikuvaa automaation teorian ja käytännön implementoinnin välille. Vaikka voisi ajatella, että esimerkiksi Azure DevOps olisi hieno ja ammattimainen työkalu sovelluksen elinkaaren hallintaan, niin todellisuudessa sen implementointiin tätä kyseistä käyttötarkoitusta varten voi mennä niin paljon aikaa, ettei se lopulta tuotakaan haluttua säästöä. HR Häirintäilmoituksen tapauksessa todennäköisesti alussa muokkauksia tulee jonkin verran, mutta sovelluksen yksinkertaisuuden vuoksi myöhemmässä vaiheessa tarve sovelluksen muutoksiin pienenee. Tällöin myös DevOpsin tai muun vastaavan pipeline-työkalun käyttöönotosta saatava hyöty jäisi todennäköisesti realisoitumatta. DevOps-pipelin avulla pystyisi automaattisesti julkaisemaan muutokset, siirtämään ratkaisun ympäristöjen välillä, lisäämään hyväksyntäprosesseja ja viemään suorituksen yhteydessä versionhallinnallisesti ratkaisu suoraan GitHubiin. (Microsoft 365 Developer 2021.)

Tuotantoon siirtoon liittyen tuli melko paljon ongelmia vastaan ja pohdin hieman, miten niiltä olisi voitu välttyä. Lisenssien puuttumisiin liittyen olisin voinut heti käyttäjän luomisen jälkeen tarkistaa, että onko lisenssit asetettu oikein, sillä minulla on näkymä asiakkaan Microsoft Endpoint Manager admin centeriin. ICT:n palvelun toimituksesta vastaava henkilö laittoi osan lisensseistä juuri ennen lomaansa ja pyysi olemaan yhteydessä kollegaani, mikäli tulee ongelmia tai tarvetta muille lisens-

seille. Toisaalta en ehtinyt edes testaamaan tuotantoon siirtoa yleisten järjestelmänvalvoja oikeuksiin liittyvän selvityksen ja siitä aiheutuneen viivästyksen vuoksi. Kyseiseen viivästykseen olisi ollut vielä hankalampi varautua aiemmin, sillä asia oli jo aiemmissa palavereissa käyty läpi HR:n kanssa eikä minulla ollut täysin varmaa tietoa, miten koko käyttöoikeusprosessi käytännössä tulee toimimaan. Samoin myös ulkopuolisen Power Apps konsultin olisi ollut hankala tuoda asiaa esille, sillä asia oli aivan aluksi käyttöoikeuspohdintoissa käyty läpi ja hän varmaan oletti asian olevan kunnossa samaan tapaan kuin itsekin. Keskustelin asiasta ICT:n kanssa ja tulin siihen tulokseen, että asiassa vaikuttaisi olevan kyse ICT:lle yleisesti tilanteesta, jossa liiketoiminta tai muu vastaava taho kuten tässä tapauksessa HR, ei ymmärrä riittävästi ympäristöjen hallintaan liittyvistä käytännöistä ja käyttöoikeuksien rajauksista.

3.8 Seurantaviikko 8 (vko 42)

Maanantai 17.10.2022

Tavoite: Saada selvitettyä oikeusasiat asiakkaan ICT:n palvelun toimituksesta vastaavan kanssa ja saada HR Häirintäilmoitus siirrettyä tuotantoon niiltä osin, kun se on tällä hetkellä mahdollista, jotta testausta tänään ja keskiviikkona olisi mahdollista aloittaa.

Toteutuminen: Ilmoitin heti aamusta ICT:n palvelutoimituksesta vastaavalle henkilölle, että huomasi viime viikolla haasteita käyttöoikeuksissa tuotantoon siirron yhteydessä. Muun muassa lisenssejä puuttui sekä Power Appsista että Power Automatesta eikä Office-lisenssiäkään ollut, jotta olisin voinut siirtää Microsoft Formsilla luodun lomakkeen tuotantoon.

Iltapäivällä pohdin HR:n toivetta kategoria kentän muuttamisesta yksivalinta-kentästä monivalintakentäksi. Kategoriolla kuvataan mihin ilmoitus liittyy ja niitä ovat kiusaaminen, epäasiallinen kohdeltu, seksuaalinen häirintä, syrjintä ja muu. Aiemmin toteutus on tehty niin, että Formsilla voi valita ainoastaan yhden kategorian. Tämä oli aluksi tehty näin muun muassa HR:n alkuperäisten toiveiden mukaan sekä noudattaen ulkopuolisen Power Apps konsultin ohjetta olla käyttämättä monivalintakenttiä, sillä niiden käsittely yksivalinta-kenttiin verrattuna on haastavampaa, ainakin Power Automateissa.

Kentän määrittäminen Formsista Dataverseen osoittautuikin haasteelliseksi ja käytin paljon aikaa asian tutkimiseen ja googlasin tilanteeseen sopivaa ratkaisua mutta sellaista ei tuntunut löytyvän. Olin jo aiemmin törmännyt monivalintakentän kohdalla haasteeseen ja sen vuoksi siirtynyt mieluummin käyttämään yksivalinta-kenttää. Huomasin testaillessa ja ratkaisua pohtiessa, että Dataversen liitin (connector) suostuu ottamaan vastaan pilkulla erotetun listan kategorioita Power Appsin puolella vastaavista valintalistan arvoista. Sain monivalinnat siten kovakoodaamalla siirret-

tyä Dataverseen. Jäljelle jäi nyt siis monivalintakentän valintojen vastaanottaminen Microsoft Formsista, konvertoiminen Dataversen valintalistan arvoja hyödyntämällä pilkulla erotetuksi merkkijonoksi ja sen syöttäminen Dataversen uuden rivin luomisen toimintoon. Tutkin ja etsin ratkaisua pidemmän aikaa mutten saanut päässä pohtimaani ratkaisua toteutettua Power Automatella. Jos ymmärrän oikein, niin minun pitäisi pystyä käymään Microsoft Formsin palauttama objekti läpi ja jokaisen kohdalla tarkistaa, että mikä valintalistan arvo vastaa mitäkin tekstiä. Sen jälkeen voisin yhdistää löydetyt numerot pilkulla erotetuksi listaksi ja Power Automaten työnkulun pitäisi toimia.

Useiden epäonnistuneiden yritysten jälkeen päädyin jättämään asian hautumaan ja päätin palata siihen myöhemmin, kunhan olen vielä varmistanut HR:ltä, että he varmasti haluavat monivalintakentän käyttöön Formsilla.

Tiistai 18.10.2022

Tavoite: Saada viimeisteltyä tuotantoon vienti. Tänään on iltapäivällä tavoitteena käydä HR:n kanssa HR Häirintäilmoituksen tilanne läpi.

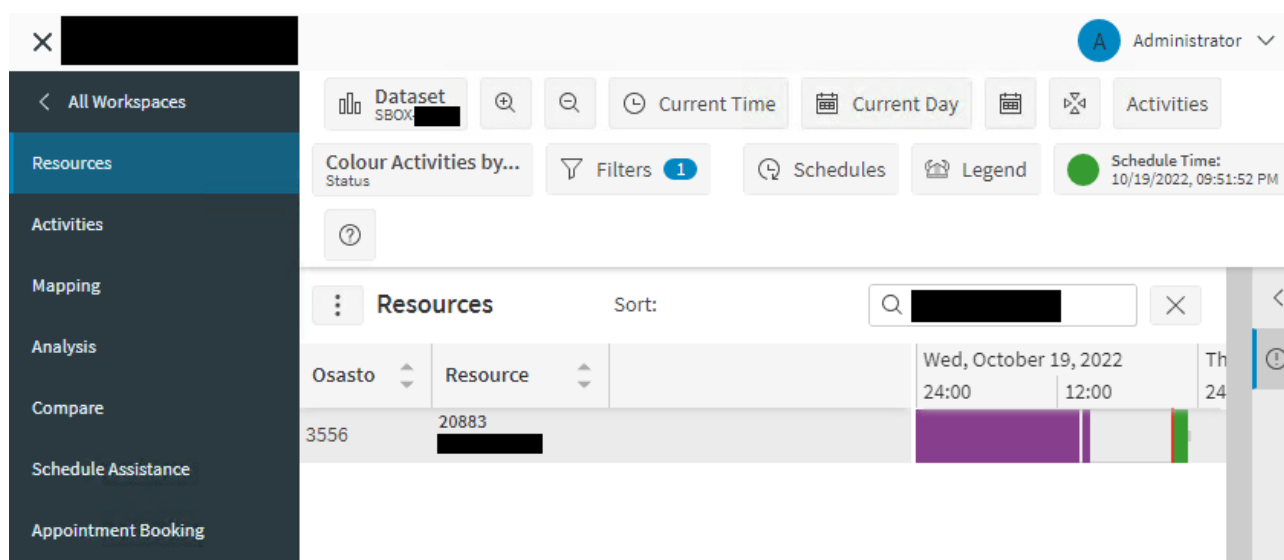
Toteutuminen: Aamupäivällä viimeistelin HR Häirintäilmoituksen tuotantoon vientiä, jonka jälkeen iltapäivällä kävimme HR:n kanssa tilanteen läpi. HR työsuhde- ja työsuojelupäällikkö ilmoitti heti palaverin aluksi, että projektiin on tulossa viivästystä ja uusi go-live aikataulu olisi 1.1.2023. Tämä oli hyvä ja helpottava uutinen sillä edellisen viikon laajavaikuttaiset häiriöt, muut tuotannon työt sekä yleisiin järjestelmänvalvojan oikeuksiin liittyvä selvitys on viivästyttänyt tuotantoon siirron aloitusta, joka puolestaan on viivästyttänyt testauksen aloitusta. Tavoitteena oli aloittaa testaus tällä viikolla mutta sovimme palaverissa testauksen aloituksen siirtämisestä marraskuun puolen välin tienoille.

Keskiviikko 19.10.2022

Tavoite: Tänään on varattuna liiketoiminnan DS-pääkäyttäjän sekä ratkaisuarkkitehdin kanssa palaveri, jossa on tavoitteena pohtia ja testata Power Apps:n soveltuvuutta DS käytössä. Olin aiemmin tehnyt testauksen omalla DS ymmärrykselläni ja todennut sen toimivaksi. Ideana on varmistaa jo testaamani idean toimivuus, saada kommentteja DS:n pääkäyttäjältä sekä avata hänelle Power Appsia enemmän.

Tälle päivälle on myös varattuna sparraussessio ulkopuolisen Power Apps konsultin kanssa, jossa on tavoitteena varmistaa omia ajatuksia tuotantoon siirtoon liittyen sekä käydä muita vastaan tulleita epäselvyyksiä ja kysymyksiä yhdessä läpi.

Toteutuminen: Pidimme testauspalaverin, jossa aluksi tuli vastaan haasteita muun muassa testidatan määrän takia. DS oli niin sanotusti tukossa aiemmista testeistä eikä halunnut kohdistaa testikäyttäjälleni uutta keikkaa, kun sen mielestä testikäyttäjälläni oli jo tarpeeksi työtilauksia ja sillä määrällä tekemistä olisi riittänyt aina perjantai-iltaan asti. Otimme pois suurimman osan testikeikoista mutta pois ottamisen jälkeen töiden väliset siirtymät aiheuttivat vielä hankaluuksia, sillä osa testikeikoista oli Turussa ja osa 3–4 tunnin matkan päässä esimerkiksi Mikkelissä, sillä testikeikkojen kohteet oli valittu summan mutikassa. Tämän takia DS ei myöskään suostunut kohdistamaan keikkaa, koska se laski, etten ehtisi työvuoron aikana suorittamaan kyseisiä keikkoja. Näiden datasta johtuvien haasteiden jälkeen keikat rupesivat kuitenkin siirtymään DS:ään ja Power Appsista tehdyt muutokset näkyivät oikein sekä ERP-järjestelmässä että DS-näkymässä. Palaverin lopputuloksena oli tyytyväinen DS-pääkäyttäjä, jonka kiinnostus Power Appsin hyödyntämiseen kasvoi entisestään. Saimme testattua DS:n toimivaksi juuri niin kuin olin ajatellut ja aiemmin testannut sen toimivan eikä vastaan tullut haasteita, joita emme palaverin aikana olisi saaneet ratkaistua.



Kuva 15. DS-näkymä testiympäristössä (Dynamic Scheduling 2022)

Kävin palaverin jälkeen ulkopuolisen Power Apps konsultin kanssa asioita läpi, ja hän selvensi minulle ALM-prosessin asioita, joista olin vielä epävarma. Opin, ettei ei-hallittuna (unmanaged) siirretty sovellus automaattisesti julkaise muutoksia vaan se täytyy tehdä tuonnin jälkeen erikseen. Hallitun ratkaisun osalta tämä tapahtuu automaattisesti. Tämän lisäksi kehitin ymmärrystäni siirron kokonaisprosessista ja huomasin, että tarvitsen myös ympäristömuuttujia, joiden avulla pystyn hallitsemaan eroavia tietoja kehitys- ja tuotantoympäristön välillä. Näistä esimerkkinä Power Automate työnkulussa käytettävä CRM-ympäristön osoite, joka on molemmille omansa. Mikäli en käytä ym-

päristömuuttujia, niin silloin jokaisen tuonnin yhteydessä nuo kohdat ylikirjoittuvat sillä mitä kehitysympäristössä on ja joudun joka kerta tuonnin jälkeen korjaamaan viittauksen oikein. Ympäristömuuttujilla pyritään siis selkeyttämään ja suoraviivaistamaan tätä prosessia.

Torstai 20.10.2022

Tavoite: Monivalintakentän ratkaisu on nyt hautunut pari päivää ja aion lähteä kokeilemaan ongelman ratkaisua uudella taktiikalla ja kirkkaammalla ajatuksella. Huomasin lisäksi tuotantoon siirron yhteydessä, että osa käyttämistäni arvoista pitäisi muuntaa joko dynaamisiksi tai ympäristömuuttujiksi, jottei jokaisen päivitykseen tai korjaukseen liittyvän tuonnin yhteydessä tarvitse tehdä paljon manuaalista työtä. Aion testata ja tutkia myös näiden käyttämistä toteutuksessani. Myös aiemmin kehitysympäristön puolelle luomani tiimit ja liiketoimintayksiköt täytyy perustaa uudestaan tuotantoympäristön puolelle.

Toteutuminen: Lähdin ensimmäisenä monistamaan kehitysympäristöstä löytyviä tiimejä ja liiketoimintayksiköitä tuotannon puolelle. Ne perustetaan Power Apps admin centeristä ja osittain niiden perustamiseen vielä hyödynnetään vanhaa Microsoft Dynamics 365:n käyttöliittymää. Esimerkiksi liiketoimintayksiköitä ei ole mahdollista tehdä Power Appsissa vaan valinta ohjaa suoraan vanhan käyttöliittymän puolelle. Tiimin voi luoda Power Appsin uudessa käyttöliittymässä ja luonnin jälkeen sitä voi muokata sekä suoraan Power Appsissa että siirtymällä Microsoft 365 Dynamicsin puolelle.

Todettuani, ettei googlaamalla löydy apua kategoria monivalintakentän arvojen syöttämiseen Power Automaten avulla Dataverseen, päätin ratkaista asian tutkimalla ja testaamalla. Maanantaina olin oppinut, että Dataverse haluaa kuvassa 16 näkyvät monivalintakentän arvot pilkulla erotettuna merkkijonona, jonka arvot tulevat suoraan kyseisen kentän numeerisista arvoista.

Display name *

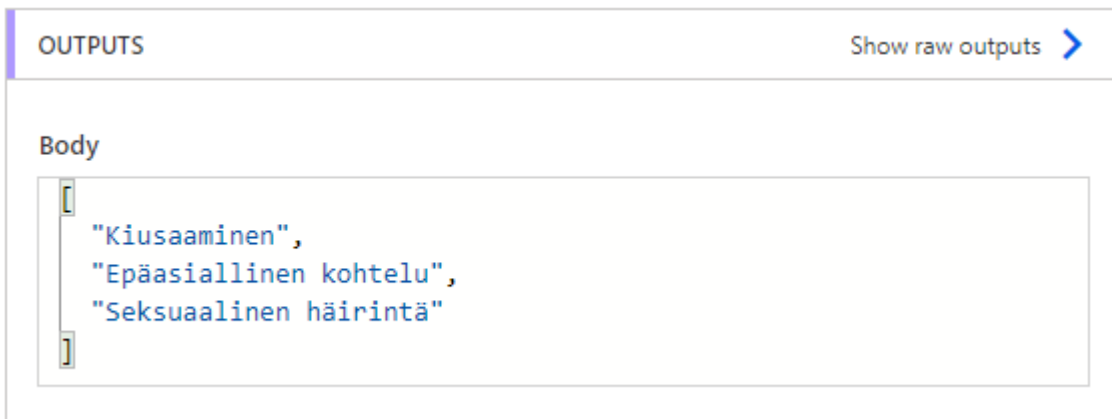
Choices

	Label *	Value *
:: <input type="checkbox"/>	Kiusaaminen	10
:: <input type="checkbox"/>	Epäasiallinen kohtelu	20
:: <input type="checkbox"/>	Seksuaalinen häirintä	30
:: <input type="checkbox"/>	Syrjintä	40
:: <input type="checkbox"/>	Muu	50

Kuva 16. Kategoria monivalintakentän numeeriset arvot (Microsoft Power Apps 2022c)

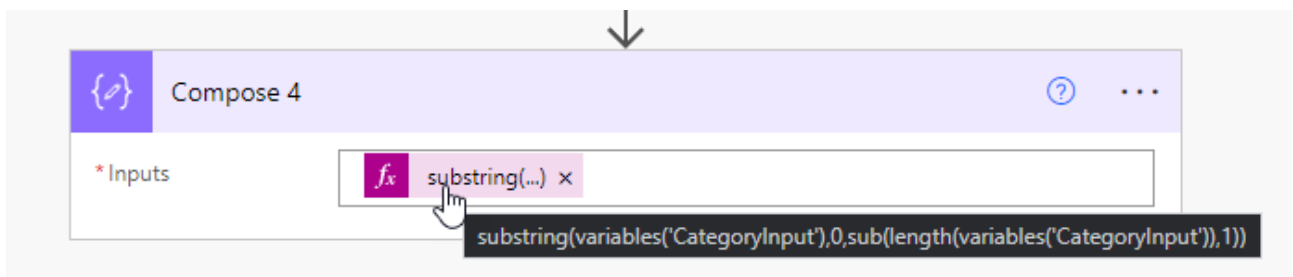
Esimerkiksi, jos käyttäjä valitsee Formsin monivalintakentästä kiusaamisen, epäasiallisen kohtelun ja syrjinnän, niin tiedän tarvittavan merkkijonosyötteen Dataversea varten olevan 10, 20, 40. Tämä oli hyvä lähtökohta ja tieto, sillä tiesin, että minun pitäisi pystyä muodostamaan Formsista tulevasta kategoria objektista edellä mainittu merkkijono.

Sain Compose- ja Parse JSON-toimintoja hyödyntäen muunnettua Formsilta tulevan vastauksen objektiksi (kuva 17).



Kuva 17. Formsin vastaus muunnettuna objektiksi (Microsoft Power Automate 2022e)

Seuraava vaihe oli käydä muunnettu objekti läpi ja tarkistaa mitä Choices-kentän numeroa kukin objektin tekstiarvo vastaa. Käytin "Apply to each" -looppia, jonka avulla pystyin käymään jokaisen objektin arvon erikseen läpi ja vertaamaan ehdon avulla, että löytyykö objektin yksittäiselle tekstile vastinetta. Jos vastine löytyy, niin sen jälkeen "Append to string variable" -toiminnon avulla lisään arvon aiemmin luotuun CategoryInput muuttujaan. Lisääminen tapahtuu if-lauseen avulla, jossa tarkistetaan jokainen item kerrallaan ja luodaan itemiä vastaava numero sekä lisätään pilkku numeron perään. Viimeisessä vaiheessa poistan vielä viimeisen pilkun CategoryInput-muuttujasta, jotta syötteestä tulee toivotunlainen.

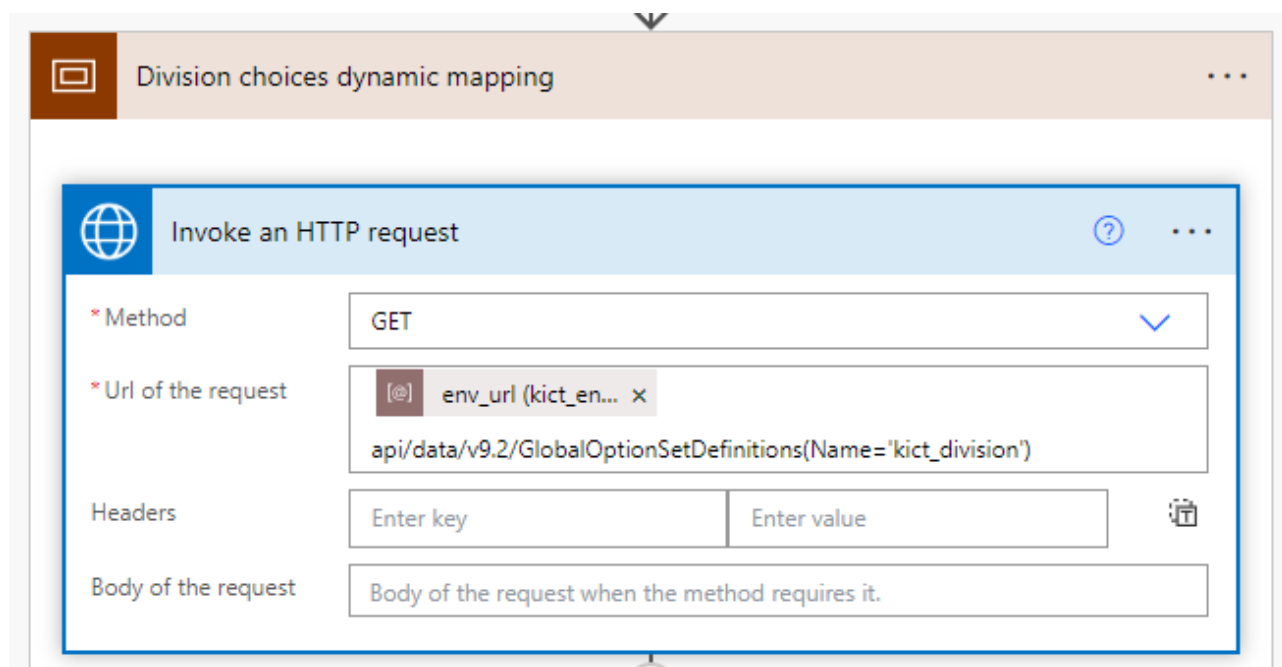


Kuva 18. Pilkun poistaminen CategoryInput-muuttujasta (Microsoft Power Automate 2022f)

Perjantai 21.10.2022

Tavoite: Eilisten onnistuneiden testailujen siivittämänä tavoitteena on tehdä tarvittavat lisäykset ratkaisun työkulkuihin. En ehtinyt eilen tutkimaan ympäristömuuttujia tarkemmin, joten se olisi myös päivän tavoitelistalla. Sovitaan lisäksi ratkaisuarkkitehdin kanssa, miten aiotaan ensi viikon POC-hankkeen palaverissa esittää asiat liiketoiminnoille, joiden on aikomus ottaa hankeen veto-vastuu jatkossa.

Toteutuminen: Tutkin heti aamusta ympäristömuuttujia. Olin jonkin verran tutustunut ympäristömuuttujiin jo videoiden avulla ja teoriatasolla mutten ollut niitä vielä aiemmin toteutuksissani konkreettisesti kokeillut. Lähdin liikkeelle selkeästä ympäristömuuttujasta, jolla aion korvata käyttämäni HTTP GET -pyynnön URL-osoitteen alkuosan (kuva 19).



Kuva 19. HTTP-pyyntö alkuosa korvattu ympäristömuuttujalla (Microsoft Power Automate 2022g)

Huomasin muutosten loppua kohden, että olin tehnyt kaikki eilisen ja tämän päivän Power Automate työkulkujen muutokset vahingossa tuotantoympäristön puolelle ja pohdin, että miten saan varmimmin siirrettyä ne kehitysympäristön puolelle, jottei ne sitten seuraavassa kehitysympäristön ratkaisupaketin tuonnissa häviäisi. Ajattelin, että varmaan tuotantoympäristöstä ratkaisun viemällä kehitysympäristön päälle homman pitäisi onnistua. Halusin kuitenkin varmistaa, että onko tässä jotain ongelmaa ja kysyin asiaa ulkopuoliselta Power Apps konsultilta. Hänen vastauksensa oli, ettei se ole normaali tapa toimia muttei äkkiseltään keksi ongelmaa tuotannosta kehitysympäristöön siirtämisessä. Vastaus sai minut kuitenkin epäröimään ja päädyin siirtämään muutokset manuaalisesti kopioimalla sillä muutoksia ei lopulta ollut kovin paljon ja sain samalla hyvän kertauksen niistä.

Viikkoanalyysi 8

Viimeistä seurantaviikkoa varjosti yksityiselämän haasteet sekä tuotannon muut tehtävät. HR Häirintäilmoituksen osalta tärkeimpänä oli saada toteutus siirrettyä kehitysympäristöstä tuotantoympäristöön. Tämä siitäkkin huolimatta, että HR ilmoitti go-live päivämäärän viivästyvän, joka toi hieman helpotusta muuten melko kiireiseen aikatauluun. POC-hankkeen osalta viikko oli enimmäkseen seuraavan viikon liiketoimintojen kanssa pidettävän jatkopalaverin odottelua ja aikaisemmin tehtyjen ratkaisujen kiillottamista sekä testaamista.

HR Häirintäilmoituksen osalta viime viikolla ilmenneiden käyttöoikeus- ja lisenssihaasteiden selvittäminen oli omalla prioriteetilla korkealla, sillä tavoitteena oli saada sovellus mahdollisimman nopeasti tuotantoon testattavaksi. Meillä oli itse asiassa sovittuna maanantaille ja keskiviikolle testausseesioita, joita ei lopulta sitten pidettykään edellä mainittujen haasteiden sekä aikataulumuutosten vuoksi. Sain kuitenkin heti alkuviikosta selvitettyä käyttöoikeushaasteet ja pääsin vihdoon siirtämään HR Häirintäilmoituksen kokonaan tuotantoon.

Tutkin myös ympäristömuuttujia, joiden avulla toteutusta saadaan tiiviimmäksi ja jotka parantavat sovelluksen siirtämistä ympäristöjen välillä vähentäen manuaalisen työn tarvetta. Ympäristömuuttajat ovat yksi osa ALM-prosessia ja niillä pyritään täydentämään sellaisia sovelluksen osia, jotka eroavat ympäristöjen välillä. (Microsoft 2022h.)

New environment variable ✕

Environment variables can have different values when re-used, enter information about this variable so that future users can understand its purpose. [Learn more](#)

Display name *

Name * ⓘ

Description

Data Type *

Choose an option ▾

- ✕ Decimal number
- { } JSON
- Text
- Yes/No
- Data source
- Secret

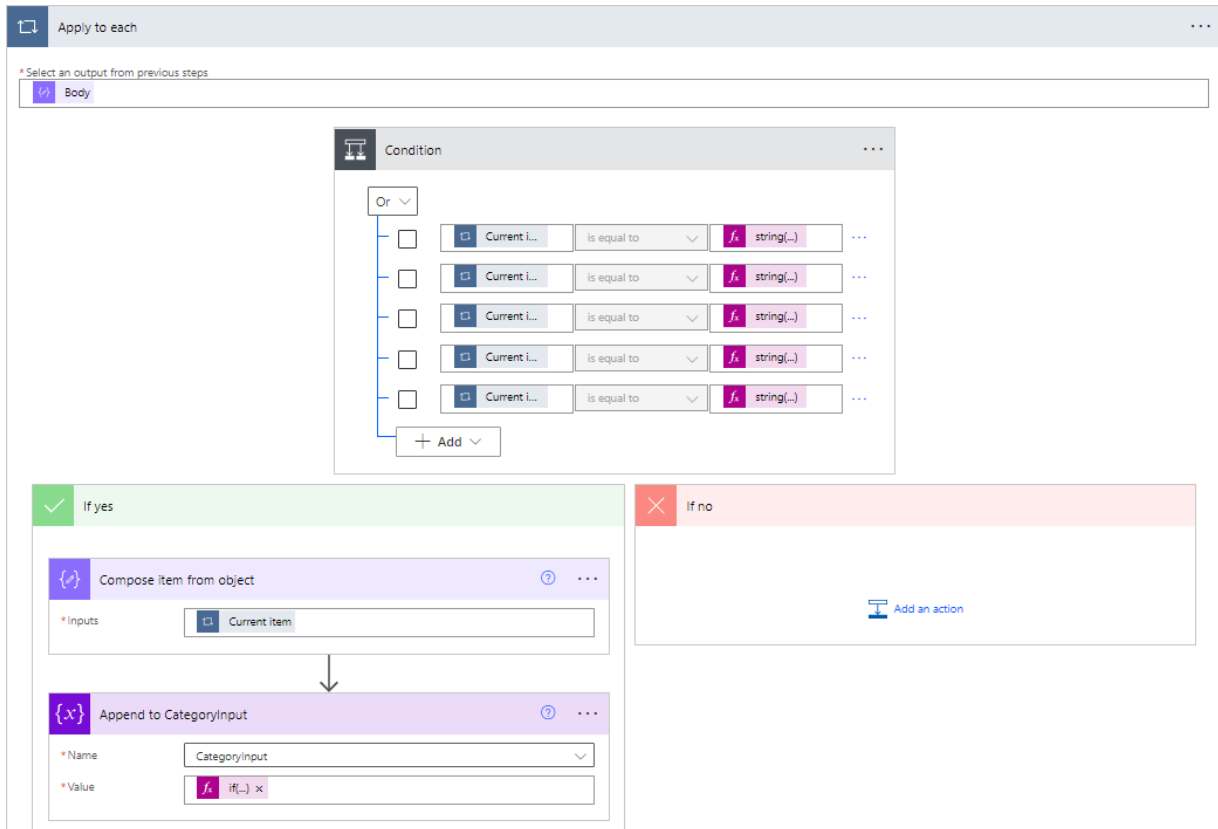
Kuva 20. Uuden ympäristömuuttujan luominen (Microsoft Power Apps 2022d)

Ympäristömuuttajat luodaan ratkaisun sisälle ja niiden luonti tapahtuu samaan tapaan kuin esimerkiksi Choice-valintalistojen tai tietokantataulujen luonti. Ympäristömuuttujalle annetaan nimi, kuvaus sekä valitaan sopiva tietotyyppi. Tällä hetkellä valittavissa on kuusi eri tietotyyppiä, jotka ovat desimaaliluku, JSON, teksti, kyllä/ei, tietolähde ja salainen koodi.

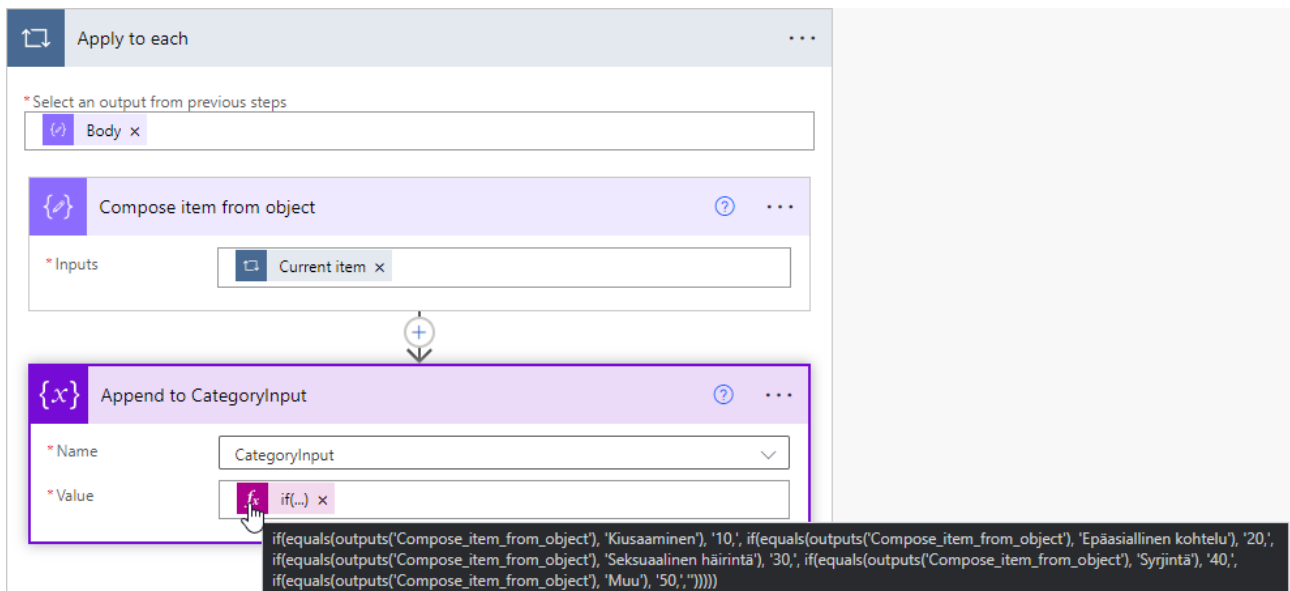
Ympäristömuuttujan nimeämisessä tulee huomioida, ettei käytä niin kutsuttuja varattuja nimiä, sillä joissain tapauksissa työnkulkua ei pysty tallentamaan. Toisekseen huonolla nimeämisellä voi vaikeuttaa omaa työtään, kun ei välttämättä erotakaan mikä ympäristömuuttuja on mihinkin tarkoitukseen. Ympäristömuuttujaan voi myös kirjoittaa tarkemman kuvauksen, josta myöhemmin selviää mihin tarkoitukseen ympäristömuuttuja on alun perin luotu. (Microsoft 2022h.)

Ympäristömuuttujia voi jopa käyttää tietolähteen määrittämiseen ympäristökohtaisesti. Tällä hetkellä Microsoftin dokumentaation mukaan tietolähteenä toimii ainoastaan Sharepoint mutta tulevaisuudessa niitä on tulossa lisää. Ensimmäisen tuotantoon siirron yhteydessä tulee huomioida, että ennen vientiä ympäristömuuttujan arvot pitää käydä poistamassa, jottei muuttujiin sidotut arvot kopioidu kehitysympäristöstä tuotantoon (Pertilä 2021).

Huomasin viikkoanalyysiä kirjoittaessa, että voisin tehdä aiemmin esitetyn kategorian tarkistuksen selkeämmin ja vähemmällä vaiheilla. Tajusin, etten tarvitse vertausehtoa ollenkaan, sillä käytännössä Formsilta ei voi tulla sellaista valintaa, jota Power Appsista ei löydy. Edellä mainittu tilanne voisi syntyä ainoastaan, jos joku loisi Formsiin uuden valinnan eikä sitä määpättäisi Power Appsin puolelle valintalistan arvoihin ollenkaan. Tämä olisi selvä logiikka- ja prosessivirhe, joten sitä ei tarvitse mielestäni tässä työnkulussa huomioida. Voin siis muokata "Apply to each" -loopin huomattavasti yksinkertaisempaan muotoon.

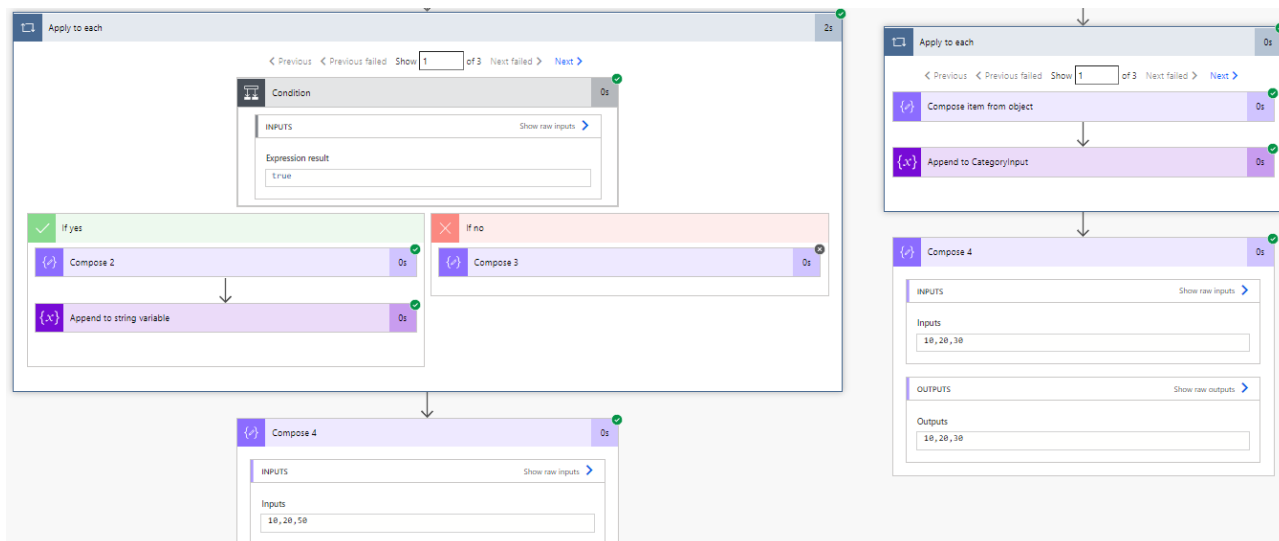


Kuva 21. Power Automate test choices työnkulku ennen muutosta (Microsoft Power Automate 2022h)



Kuva 22. Power Automate Test Choices työnkulku muutoksen jälkeen (Microsoft Power Automate 2022i)

Muutoksen jälkeen looppi on paljon selkeämpi ja suoritus on myös huomattavasti nopeampi, kun joka välissä ei tarvitse tehdä tarkistusta ehtolauseetta käyttäen. Aiemmassa suorituksessa vanhalla tavalla meni muutama sekunti ja nyt uudella muutetulla tavalla suorituksessa menee alle sekunti.



Kuva 23. Test Choices toteutuksen erot - vasemmalla vanha, oikealla uusi (Microsoft Power Automate 2022j)

Ensi viikon maanantaina POCista on jatkopalaveri liiketoimintojen kanssa, jossa käydään nykytilanne läpi ja pohditaan yhdessä suuntaviivoja jatkon osalta. Ennen palaveria olisi hyvä saada testattua nykytoteutus omalla käyttäjälläni tuotannon tapausta vastaavalla tavalla. Testauksen avulla saisin tärkeää tietoa sovelluksen valmiudesta kehitysympäristössä tapahtuvaan pilotointiin, jonka tulevat suorittamaan asiakkaan ERP-järjestelmän pääkäyttäjät ja mahdollisesti tuotannon kehitystiimin työntekijät.

Samoin testausta on luvassa myös HR Häirintäilmoituksen osalta, sillä lupasin HR:lle että marraskuun puolen välin testejä varten HR Häirintäilmoituksen pitäisi olla sellaisessa kunnossa, että voidaan tehdä HRIS-spesialistin kanssa tuotannon varmistustestaus läpi ongelmitta. Nyt kun olen saanut ratkaisun siirrettyä tuotantoon ja minulla on suurin piirtein valmis prosessi ratkaisun viemisestä kehitysympäristöstä tuotantoon, testauksen tekemisen pitäisi onnistua helposti ja mahdollisten muutosten pitäisi siirtyä kätevästi ympäristöjen välillä.

4 Pohdinta

Opinnäytetyöprosessin aikana tuli monenlaisia yllätyksiä ja haasteita vastaan ja ensimmäiset haasteet tulivat jo aivan opinnäytetyöprosessin alkumetreillä. Olin suunnitellut aloittavani opinnäytetyön tekemisen jo ennen kesälomia ja näin myös teinkin mutta huomasin nopeasti, että ensimmäisestä aiheesta olisi ollut todella vaikea saada aikaiseksi järkevää ja kiinnostavaa sisältöä. Aloitus tuli aivan liian nopeasti, joten en ollut ehtinyt tekemään kunnon suunnitelmaa. Lisäksi aihe oli itselle täysin uusi, joten se olisi vaatinut todella paljon perehtymistä ennen minkäänlaisen järkevän sisällön kirjoittamista. Perehtymisen oli ajatus tapahtua työajalla työtehtävien niin salliessa. En kuitenkaan työkiireiden vuoksi ehtinyt perehtymään aiheeseen päivisin, joten perehtyminen jäi iltaan ja kirjoittaminen meni ja olisi jatkossakin mennyt pitkälti yötöiksi. Ensimmäisen seurantaviikon viikkoanalyysin palautteen jälkeen tulimme yhdessä ohjaajan kanssa tulokseen, ettei työtä kannata sellaiseenaan jatkaa. Sovimme, että on parempi pohtia uutta aihetta ja katsoa asiaa uudestaan syksyllä lomien jälkeen.

Olen todella kiitollinen, etten vain jatkanut ensin valitsemani aiheen kanssa, vaikka aluksi ajatus opinnäytetyön keskeyttämisestä tuntuikin hankalalta. Opinnäytetyöprosessi olisi ollut todennäköisesti todella haastava, jos olisin jatkanut ensin valitsemani aiheen kanssa. Varsinkin sisältö ja aika-taulu olisi luonut isoja haasteita prosessiin. Tämä uusi opinnäytetyön aihe valikoitui pidemmän pohdinnan tuloksena oman kiinnostuksen, tulevaisuuden hyödyn sekä edellä mainitun oppimiskokemuksen kautta. Ymmärsin, ettei päiväkirjaopinnäytetyön aihe voi olla oma, irrallinen kokonaisuutensa, vaan sen tulee liittyä tiiviisti päivittäiseen tekemiseen. Power Platform ja Power Apps ovat varmasti tulevaisuudessakin kysytyjä ja kuten Gartnerin (Wong ym. 2022) raportistakin käy ilmi, sovelluskehitys tulee siirtymään low-code-kehityksen puolelle sen tarjoamien liiketoimintamahdollisuuksien sekä kustannusten leikkausten ansiosta. Yksi perustelu aiheen valintaan löytyykin taustalla kehittyneestä tavoitteesta kehittää omaa Power Apps osaamista siten, että pystyn tekemään Microsoftin Learnin kautta Power Appsiin liittyviä sertifikaatteja. Sertifikaattien sekä projektien tuostosten avulla pystyn jatkossa vakuuttamaan asiakkaat ja muut tahot Power Apps osaamisestani. Aiheen valinta on siis samalla ollut myös panostus tulevaisuuteen, ja ajatus on välillä käynyt jo useamman vuoden päässä pohtimassa mihin kaikkeen uutta osaamista voisi hyödyntää.

Kesälomien jälkeen lähtökohta opinnäytetyön tekemiseen oli mielenkiintoinen, sillä toimin opinnäytetyön aiheeseen liittyvissä kahdessa Power Apps asiakasprojektissa kehittäjänä, vaikka perehtymisen kyseiseen low-code-teknoologiaan olin aloittanut vasta viime vuoden loppupuolella. Ennen Power Appsia minulla oli vain hieman aiempaa kokemusta low-code-sovelluskehityksestä Oracle Apexin ja Outsystems alustojen kautta ja ymmärsin perusteita Javasta, JavaScriptistä, Reactista sekä muista opiskeluissa käytetyistä ohjelmointiteknoologioista. Nämä kaksi asiakasprojektia olivat

kuitenkin ensimmäiset kunnolliset sovelluskehitykset, joissa toimin kehittäjän roolissa asiakasprojekteissa. Aluksi epäilin riittääkö minulla osaaminen kehittämään asiakkaan tarpeet ja toiveet täyttävät sovellukset. Olin lähtötilanteessa arvioinut olevani aloitteleva toimija, joka tarvitsi vielä apua sekä videoiden että ulkopuolisen konsultin kautta. Epäilykset osaamisesta nousivat ajoittain viikkoanalyysijä pohtiessani pintaan, mutta huomasin opinnäytetyöprosessin edetessä, että pysyin tehtävieni tasalla eikä asiakas missään vaiheessa vaikuttanut epäilevän osaamistani.

Jo alussa oli selvää, että opinnäytetyön osaamisen kehittämisen tavoitteiden osalta työn painopisteen täytyi keskittyä nimenomaan opinnäytetyön aiheisiin. Onneksi myös asiakkaan pääkontaktina toimiva business partner oli samaa mieltä asiasta. Työn painopisteen keskittymisessä Power Appsiin oli myös heille selvää etua, sillä kehittäminen onnistui joustavammin ja sovellusten kanssa tuli jatkuvaa edistymistä. Joka viikko tiiviisti Power Appsia ja muita siihen liittyviä teknologioita käytäessä huomasin selvää kehittymistä ja sain lähes jokaiselle seurantaviikolle oman käsiteltävän teeman, jonka kautta pääsin pohtimaan ja analysoimaan syvällisemmin mistä viikossa on ollut kyse.

Olen mielestäni kehittynyt opinnäytetyöprosessin aikana hyvin asettamissani tavoitteissa. Eniten teknisissä, sillä seurantajakson aikana tekeminen painottui tekniseen puoleen mutta olen myös opinut paremmin ymmärtämään liiketoiminnan tarpeita ja priorisoimaan niitä järkevästi. Tämän lisäksi olen löytänyt uusia toimintamalleja pohdintojen lomassa. Huomasin, että etenkin viikkoanalyysistä oli suurta apua oman oppimisen analysoimisessa, sillä niiden kautta pääsin syventymään viikon tapahtumiin ja sitä kautta löytämään uusia ratkaisumalleja ja metodeja, joilla asiat voisi tehdä paremmin. Opin muun muassa versionhallinnasta kokeilemalla ja tutustumalla, ettei sitä kannatta vielä käyttää Power Appsin kanssa tuotantoympäristössä. GitHubin yhdistäminen pahimmillaan kadotti tekemiäni muutoksia ja jouduin myös joka kerta erikseen kirjautumaan siihen. ALM:stä eli sovelluksen elinkaaren hallinnasta opin lisää muun muassa siitä, minkälaisia vaihtoehtoja ALM:n osalta on olemassa. Power Appsiin liittyen opin ALM-prosesseista muun muassa sovelluksen ympäristöjen välisestä siirtämisestä, milloin automatisointia kannattaa tehdä ja milloin kannattaa tyytyä kevyempään hallintamalliin sekä monenlaisia tapoja siirron jälkeisten toimenpiteiden vähentämiseen. Tällaisia ovat muun muassa ympäristömuuttajat ja dynaamisesti haetut tiedot työkuluissa.

Huomasin omia päiväkirjamerkintöjä analysoidessa ja mennyttä viikkoa pohtiessa, että välillä on hyvä antaa asian vain olla, jos siihen ei ole löytynyt järkevässä ajassa ratkaisua. Useimmiten näissä tapauksissa hetken huili teki hyvää ja löysin seuraavalla kerralla samaa ongelmaa tutkiessa ratkaisun nopeasti ja yllätyin kuinka helppo ja selkeä ratkaisu lopulta oli. Samaan tapaan huomasin

myös, etten välillä ongelmanratkaisussa keskittynyt heti kaikista olennaisempaan eli ongelman tunnistamiseen. Olen urani aikana tehnyt paljon ongelmanratkaisua ja useimmiten vastaan tulleissa tilanteissa virheilmoituksen tarkempi analysointi on vienyt ongelmanratkaisussa eteenpäin. Huomasin saman päteväni myös sovelluskehityksen osalta mutten kuitenkaan joka kerta tajunnut siitä aloittaa. Asiaa tarkemmin miettiessä ja tutkiessa löysin netistä George Polyan 1940-luvulla kehittelemän nelivaiheisen ongelmanratkaisumallin. Siinä ongelma ensin tunnistetaan, jonka jälkeen laaditaan suunnitelma, jolla ongelmaa lähdetään ratkomaan. Kokeilin kyseistä metodia heti seuraavan kerran, kun sopiva tilanne ilmaantui ja huomasin siitä olevan paljon hyötyä. Ongelman tunnistaminen ja ongelmaan liittyvien asioiden auki kirjoittaminen auttoi hahmottamaan ongelmaa aivan uudella tavalla. Polyan malli tulee jäämään omaan työkalupakkiin ja aion käyttää sitä jatkossakin sen sijaan, että vain suoraan aloitan ongelman selvittämisen ilman suunnittelua. (Kanna 2021.)

Erilaisten teknologioiden ymmärtämisessä otin ison harppauksen kohti itsenäistä kehittämistä. Siinä missä opinnäytetyön alussa olin vielä melko epävarma Power Appsin, Power Automaten ja Power BI:n kehittämisestä enkä ollut Azurestakaan ymmärtänyt muuta kuin perusidean, nyt opinnäytetyöprosessin viime metreillä tuntuma osaamiseen on aivan eri tasolla. Pystyn itsenäisesti tuottamaan tarvittavia toiminnallisuuksia sovelluksiin, pystyn luomaan toimivan käyttöliittymän Power Appsiin ja tukemaan sovelluksia Power Automaten työnkulkujen avulla. Tunnen etenkin Power Appsiin liittyviä parhaita käytäntöjä ja pystyn perustellen kertomaan asiakkaalle erilaisista vaihtoehdoista toteutuksen tekemiseen. Opinnäytetyöprosessin aikana tutustuin itsenäisesti myös Azuren Logic Appsiin, jonka huomasin olevan todella samanlainen, kuin Power Automaten. Pystyin muun muassa opinnäytetyöprosessin keskivaiheilla ja etenkin loppuvaiheessa muokkaamaan aiemmin luotuja integraatioita toimivammiksi ja luomaan integraatioita lähes alusta asti.

Integraatioiden kehitykseen liittyen ainoastaan integraation luominen jäi opinnäytetyöprosessin aikana tekemättä. Integraation luomisen, ylläpidon sekä yleisesti Power Appsiin liittyvän lisensoinnin osalta minulla on vielä opittavaa ja ne voisivatkin luonnollisesti olla seuraavia kehittämiskohteita oman osaamiseni osalta. Olemme jo tunnistanee, että POC-hankkeeseen tarvitaan useita integraatioita lisää. Tämän lisäksi myös lisensointia ja sovelluksen mahdollista ylläpitoa täytyy ruveta pohtimaan, jos päädytään siihen, että POC-hankkeen tuotoksella voidaan korvata nykyinen asiakkaan käyttämä mobiilisovellus.

Yhden yksittäisen Power Automateen liittyvän oivalluksen nostan isoon arvoon, sillä siihen en löytänyt sopivaa ratkaisua googlaamalla, YouTube-videoita katsomalla tai ulkopuolisen konsultin kanssa keskustelemalla. Kyseisessä tapauksessa piti saada mäpättyä Microsoft Formsilta tulevat monivalintakentän vastaukset Power Automaten kautta Dataversen tietokantaan. Tämä osoittautui

hankalaksi ja aikaa vieväksi selvitykseksi. Ulkopuolinen Power Apps konsulttikin oli jo alussa varoittanut monivalintakentän käytön haasteista ja suositellut niiden välttämistä, jos se vain on mahdollista. Tässä tapauksessa se ei kuitenkaan ollut mahdollista, sillä HR näki monivalintavastausvaihtoehdot tärkeinä loppukäyttäjien kannalta. Lopulta ratkaisun löydyttyä ihmettelin sen helppoutta ja sitä, miten en vimmatusta googlaamisesta huolimatta löytänyt vastausta netin syövereistä. Ensimmäinen ongelma oli keksiä missä muodossa Dataverse suostuu ottamaan vastaan monivalintakentän Formsilta tekstinä tulevat vastaukset. Sen jälkeen piti selvittää, miten saan muutettua tekstivastaukset kyseiseen Dataversen hyväksymään muotoon. Loppujen lopuksi vaadittiin vain merkkijono, jossa oli pilkulla eroteltuna tekstivastauksia vastaavat Dataversen monivalintakentän numeeriset arvot. Jos pitäisin Power Appsiin liittyvää blogia tai YouTube-kanavaa, niin tästä tekisin ehdottomasti ensimmäisen postaukseni, jotta voisin säästää muiden aikaa näinkin yksinkertaisen ratkaisun selvittämiseltä.

Opinnäytetyön alkupuoliskolla huomasin myös tarpeen päivittäiselle läpikäynnille epäselvyyksien ja kommunikaatiohaasteiden takia. Käynnistin päivittäispalaverit, joissa tavoitteena oli käydä päivän tehtäviä, mahdollisia päivän polttavia kysymyksiä ja ongelmatilanteita läpi. Tällä hain sitä, että jokainen hankkeeseen osallistuva henkilö olisi tietoinen missä hankkeen osalta mennään, mitä pitäisi tehdä ja mihin menessä. Päivittäispalaverien käynnistäminen liittyi myös alussa määrittelemääni tavoitteeseen ketterän kehityksen ymmärryksen kasvattamisesta. Harmillisesti opinnäytetyön aikana kummassakaan projektissa ei kuitenkaan ehtinyt kunnolla tulla sellaisia piirteitä, joiden kautta olisin päässyt syvällisemmin kasvattamaan ymmärrystäni ketterästä kehityksestä. Arvioisin sen johtuvan nimenomaan hankkeen POC-vaiheesta ja siitä, että asiakkaan puolelta liiketoiminta tuli mukaan vasta seurantaviikkojen loppupuolella. He esimerkiksi nimittivät vasta viimeisen seurantaviikon aikana POC-hankkeelle tuoteomistajan. HR Häirintäilmoituksen osalta tekeminen oli koko ajan melko ketterää mutta aktiivisesti sen edistämiseen osallistui vain muutama henkilö. Kummassakaan ei mielestäni varsinaisesti voida puhua Scrumista, vaikka molemmat tehtävälistöineen ja läpikäynteineen kovasti Scrumia muistuttavatkin. POC-hankkeen osalta koordinointi tapahtui aluksi minun ja ratkaisuarkkitehdin toimesta ja ainoastaan isommat suuntaviivat saatiin ICT:n kanssa käydystä seurantapalavereista.

Yllätyin käyttöoikeuksiin liittyvien haasteiden määrästä. Ensimmäiset haasteet tulivat vastaan heti alkupuoliskolla, kun meillä ei aluksi ollut riittäviä oikeuksia integraatioiden muokkaamiseen. Käyttöoikeudet olivat läsnä monena seurantaviikkona opinnäytetyöprosessin aikana. Välillä ongelmat liittyivät puuttuviin oikeuksiin, välillä oikeuksien laatuun ja välillä käyttöoikeusprosessin kankeuteen. Kehityksen sujuvuuden kannalta olisi ehdottoman tärkeää, että kehittäjillä on tarvittavat oikeudet tai jos ne puuttuvat, niin ne voitaisiin lisätä viiveettä ilman hankalia prosesseja.

Suorituskyvyn osalta olin valmistautunut kompromisseihin sillä aiempi ymmärrykseni oli, että low-code-alustoilla tehdessä suorituskyky saattaa kärsiä. Tämän kyseisen tiedon myötä tein jo kehitysvaiheessa alustavaa suorituskykytestausta, jonka myös raportoin viikkoanalyysin yhtenä teemana. Suorituskykytestauksen tavoitteeksi asetin, ettei sovellus saisi jäädä liian kauan ajaksi lataamaan, jotta sovellusta voitaisiin oikeasti hyödyntää liiketoimintakriittisessä mobiilisovelluksessa työntekijöiden käyttämänä. Pystyin jo ensimmäisen suorituskykytestauksen pohjalta vaikuttamaan sovelluksen nopeuteen jonkin verran ja opin, että on paljon erilaisia vaihtoehtoja sovelluksen suorituskyvyn parantamiseksi. Löysin niitä sekä Microsoftin dokumentaation, YouTube-videoiden että sovelluksen omien analyysitoimintojen avulla.

Minulle juolahti jossain vaiheessa opinnäytetyöprosessia mieleen, että voisin toteuttaa Power Appsilla ratkaisukannan, josta voisin hakea aiemmin ratkaisemieni tapausten yksityiskohtia ja ohjeita. Olen tätä aihetta pohtinut aiemminkin ja olen myös jonkinlaisia Excelillä tehtyjä virityksiä tehnyt, mutta nyt varsinkin opinnäytetyön loppupuolella itseluottamus Power Apps tekemiseen on kasvanut merkittävästi, että ratkaisukannan pystyttäminen olisi helppo juttu. Sen avulla voisin parantaa asiakkaalle tarjoamani palvelun laatua ja auttaa uusiutuviissa ongelmissa entistä nopeammin.

Opin myös prosessin aikana paljon Microsoftin tulevaisuuden suunnitelmista etenkin Power Appsin ja low-coden osalta. Yksi heidän suunnitelmistansa on siirtyä imperatiivisesta sovelluskehityksestä deklaratiiiviseen sovelluskehitykseen ja tämä muutos on jo aloitettu. Vaikka olen tietämättä tehnyt imperatiivista ohjelmointia, jossa ohjelmalle kerrotaan mitä halutaan, miten halutaan ja miten lopputulokseen päästään – kuulostaa deklaratiivinen jo ensi kuulemalta huomattavasti paremmalta, kun siinä ohjelmalle tarvitsee enää kertoa mitä halutaan.

Vaikka Microsoftin ja muiden isojen low-code-sovelluskehitykseen satsaavien tahojen kaunis ajatus ja visio on, että sovelluskehitystä voisi tehdä kuka vaan, mahdollisesti ilman minkäänlaista ohjelmointiosaamista, niin siellä asti ei vielä olla. Se ettei low-code ole vielä valmis siihen ei sinällään yllättänyt, sillä olen työurani aikana nähnyt, vaikka ja minkä tasoisia liiketoiminnan käyttäjiä ja nyt viimeisimpinä työ- ja opiskeluvuosina ohjelmointiin tutustuneena ymmärrän molempia puolia. Monia aiemmin aina uudestaan ja uudestaan tehtäviä asioita on pystytty vakioimaan ja tuomaan omina kokonaisuuksina käytettäväksi low-code-alustoille. Pystytäänkö kuitenkaan kaikkea korvaamaan low-code-teknologioilla vai meneekö low-code-alustoja hyödyntävä sovelluskehittäminen jossain vaiheessa jo liian hankalaksi? Milloin Excelin kaltaisten monimutkaisempien funktioiden ja tavallisten ohjelmakoodien raja hämärtyy? Edellä mainittuja kysymyksiä voisi tulevaisuudessa tutkia enemmän ja sitä kautta kehittää low-code-alustojen mahdollisuuksia pärjätä monimutkaisempien kehitysten tekemisessä perinteiseen ohjelmointiin ja sovelluskehitykseen verrattuna.

Lähteet

Bergius, K. 2014. Mikä se Azure oikein on? Luettavissa: <https://sulava.com/pilvi-infrastruktuuri/mika-se-azure-oikein/>. Luettu: 23.11.2022.

Berloty, P. 2021. 7 keys to creating the wow effect in a demo. Luettavissa: <https://en.modjo.ai/blog/7-keys-to-creating-the-wow-effect-in-a-demo>. Luettu: 1.10.2022.

Bradley, S. 2020. Do you trust your admins? 5 tips to manage administrator access. Luettavissa: <https://www.csoonline.com/article/3529993/do-you-trust-your-admins-5-tips-to-manage-administrator-access.html>. Luettu: 26.9.2022.

Cocca, G. 2022. What is Debugging? How to Debug Your Code for Beginners. Luettavissa: <https://www.freecodecamp.org/news/what-is-debugging-how-to-debug-code/>. Luettu: 10.9.2022.

Cohen, N. 2022. Performance Testing vs. Load Testing vs. Stress testing. Luettavissa: <https://www.blazemeter.com/blog/performance-testing-vs-load-testing-vs-stress-testing>. Luettu: 10.10.2022.

Delano, L. 2018. New App checker helps you fix errors and make accessible apps. Luettavissa: <https://powerapps.microsoft.com/en-us/blog/new-app-checker-helps-you-fix-errors-and-make-accessible-apps/>. Luettu: 26.9.2022.

Dell 2019. End Users are Speaking: It's IT's Turn to Listen. Luettavissa: <https://www.computerworld.com/article/3331170/end-users-are-speaking-it-s-it-s-turn-to-listen.html>. Luettu: 11.10.2022.

Devaney, M. 2022. 3 Simple Ways To Change The Owner Of A Power Apps App. Luettavissa: <https://www.matthewdevaney.com/3-simple-ways-to-change-the-owner-of-a-power-apps-app/>. Luettu: 15.10.2022.

Dorrani, R. 15.6.2021a. Create push notifications for Power Apps mobile – Tutorial. Video. Katsottavissa: <https://www.youtube.com/watch?v=YLYsognPA6s>. Katsottu: 8.9.2022.

Dorrani, R. 23.2.2021b. Power Apps Performance Optimization Tips. Video. Katsottavissa: https://www.youtube.com/watch?v=_2jDGqtqqi8. Katsottu: 15.9.2022.

Dorrani, R. 1.11.2021c. Start using these 7 Power Apps Formulas (Power Fx). Video. Katsottavissa: <https://www.youtube.com/watch?v=IW6BT9rsQ7g>. Katsottu: 15.9.2022.

Dorrani, R. 11.7.2022. Solutions in Power Platform | ALM, Power Apps, flows & Environments. Video. Katsottavissa: <https://www.youtube.com/watch?v=Xo-TvZ9N3BM>. Katsottu: 15.10.2022.

Drumond, C. s.a. Is the Agile Manifesto still a thing? Luettavissa: <https://www.atlassian.com/agile/manifesto>. Luettu: 30.11.2022.

Dunn, P. 2018. PowerApps Canvas App Coding Standards and Guidelines. Luettavissa: <https://powerapps.microsoft.com/en-us/blog/powerapps-canvas-app-coding-standards-and-guidelines/>. Luettu: 12.9.2022.

Dunnam, A. 2022. Change Power Apps Owner with Power Automate in Under 10 Minutes. Video. Katsottavissa: <https://www.youtube.com/watch?v=DgBIVuf9Hel>. Katsottu: 16.10.2022.

Dynamic Scheduling 2022. Kuvakaappaus Dynamic Scheduling.

Fisette, L. 2021. When Is Your App Too Complex for Power Apps? Luettavissa: <https://corebits.com/blog/when-is-app-too-complex-for-power-apps/>. Luettu: 12.10.2022.

Geeksforgeeks 2022. Debugging: Tips To Get Better At It. Luettavissa: <https://www.geeksforgeeks.org/debugging-tips-to-get-better-at-it/>. Luettu: 20.9.2022.

Hamel, G., Doz, Y. & Prahalad, C.K. 1989. Collaborate with Your Competitors – and Win. Luettavissa: <https://acervo-digital.espm.br/Artigos/ART/2017/374195.pdf>. Luettu: 11.10.2022.

Harmon, S. 2019. To Meet Or Email -- That Is The Question. Luettavissa: <https://www.forbes.com/sites/harmoncullinan/2019/05/16/to-meet-or-email-that-is-the-question/?sh=5a9ea08050dc>. Luettu: 4.9.2022.

Hart, M. 2021. How to Deliver the Perfect Product Demo. Luettavissa: <https://blog.hubspot.com/sales/product-demo>. Luettu: 4.10.2022.

Heaney, A. 2020. Introducing Test Studio to build end-to-end tests and maintain your app quality. Luettavissa: <https://powerapps.microsoft.com/en-us/blog/introducing-test-studio-to-build-end-to-end-tests-and-maintain-your-app-quality/>. Luettu: 20.9.2022.

Heusser, M. 2019. Debugging. Luettavissa: <https://www.techtarget.com/searchsoftwarequality/definition/debugging>. Luettu: 23.11.2022.

Ihana, S. 2017. 5 Things How to Improve IT Communication for End Users. Luettavissa: <https://www.linkedin.com/pulse/5-things-how-improve-your-communication-end-users-sari-aliu/>. Luettu: 11.10.2022.

Indeed editorial team 2021. Power Interest Grid: Definition and How To Use. Luettavissa: <https://www.indeed.com/career-advice/career-development/power-interest-grid>. Luettu: 12.9.2022.

json.org s.a. Introducing JSON. Luettavissa: <https://www.json.org/json-en.html>. Luettu: 23.11.2022.

Kanna, M. 2021. How to Solve Coding Problems with a Simple Four Step Method. Luettavissa: <https://www.freecodecamp.org/news/how-to-solve-coding-problems/>. Luettu: 10.9.2022.

Keating, G. 2021. Data Mapping 101: What It Means and How to Do It. Luettavissa: <https://segment.com/blog/data-mapping/>. Luettu: 30.11.2022.

Kothadiya, A. s.a. Mistakes I've Made When Giving Product Demos. Luettavissa: <https://www.avoma.com/blog/sales-product-demos-mistakes>. Luettu: 4.10.2022.

Kubrak, S. 2021. Technical Due Diligence: Meaning, Process and Checklist. Luettavissa: <https://djangostars.com/blog/tech-due-diligence/>. Luettu: 11.10.2022.

Li, M. 2019. Performance testing & stress testing tools for PowerApps. Luettavissa: <https://powerusers.microsoft.com/t5/Building-Power-Apps/Performance-testing-amp-stress-testing-tools-for-PowerApps/td-p/407100>. Luettu: 10.10.2022.

Lifesize 2021. The Top 7 Tips for Pulling Off a Great Demo Presentation. Luettavissa: <https://www.lifesize.com/en/blog/demo-presentation/>. Luettu: 4.10.2022.

Lindhorst, G. 2021a. App.StartScreen: a new declarative alternative to Navigate in App.OnStart. Luettavissa: <https://powerapps.microsoft.com/en-us/blog/app-startscreen-a-new-declarative-alternative-to-navigate-in-app-onstart/>. Luettu: 22.9.2022.

Lindhorst, G. 2021b. Power Fx: Introducing Named Formulas. Luettavissa: <https://powerapps.microsoft.com/en-us/blog/power-fx-introducing-named-formulas/>. Luettu: 22.9.2022.

Liu, H. 2021. Automated Testing with Logic Apps Standard. Luettavissa: <https://techcommunity.microsoft.com/t5/integrations-on-azure-blog/automated-testing-with-logic-apps-standard/ba-p/2960623>. Luettu: 20.9.2022.

Lowery, L. 2020. How to prepare the perfect product demo. Luettavissa: <https://lukelowrey.com/product-demo-checklist/>. Luettu: 2.10.2022.

Malsam, W. 2021. What Is Proof of Concept (POC)? Definition, Steps & Best Practices. Luettavissa: <https://www.projectmanager.com/blog/proof-of-concept-definition>. Luettu: 27.9.2022.

Marchewka, J. 2014. Information technology project management. 4. painos. John Wiley & Sons, Inc. Hoboken.

Mathenge, J. & Stevens-Hall, J. The Seven ITIL 4 Guiding Principles. Luettavissa: <https://www.bmc.com/blogs/itil-guiding-principles/>. Luettu: 5.9.2022.

Microsoft 2022a. Troubleshoot and diagnose workflow failures in Azure Logic Apps. Luettavissa: <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-diagnosing-failures?tabs=consumption>. Luettu: 11.9.2022.

Microsoft 2022b. Power Automate vs Logic Apps. Luettavissa: <https://docs.microsoft.com/en-us/microsoft-365/community/power-automate-vs-logic-apps#learning-azure-logic-apps>. Luettu: 10.9.2022.

Microsoft 2022c. Tips and best practices to improve performance of canvas apps. Luettavissa: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/performance-tips>. Luettu: 25.9.2022.

Microsoft 2022d. Advanced monitoring concepts. Luettavissa: <https://learn.microsoft.com/en-us/power-apps/maker/monitor-advanced#unsupported-scenarios-for-monitor>. Luettu: 22.9.2022.

Microsoft 2022e. About admin roles in the Microsoft 365 admin center. Luettavissa: <https://learn.microsoft.com/en-us/microsoft-365/admin/add-users/about-admin-roles?view=o365-worldwide>. Luettu: 26.9.2022.

Microsoft 2022f. Overview of application lifecycle management with Microsoft Power Platform. Luettavissa: <https://learn.microsoft.com/en-us/power-platform/alm/overview-alm>. Luettu: 16.10.2022.

Microsoft 2022g. Language function in Power Apps. Luettavissa: <https://learn.microsoft.com/en-us/power-platform/power-fx/reference/function-language>. Luettu: 8.10.2022.

Microsoft 2022h. Environment variables overview. Luettavissa: <https://learn.microsoft.com/en-us/power-apps/maker/data-platform/environmentvariables>. Luettu: 23.10.2022.

Microsoft 2022i. XML:n perusteet. Luettavissa: <https://support.microsoft.com/fi-fi/office/xml-n-perusteet-a87d234d-4c2e-4409-9cbc-45e4eb857d44>. Luettu: 23.11.2022.

Microsoft 2022j. Azure Data Factory. Luettavissa: <https://azure.microsoft.com/en-us/products/data-factory/#overview>. Luettu: 23.11.2022.

Microsoft 365 Developer 2021. Getting started with Power Apps ALM. Video. Katsottavissa: https://www.youtube.com/watch?v=_jeFTasxYNk. Katsottu: 14.10.2022.

Microsoft Excel 2022a – 2022c. Kuvakaappaus Microsoft Excel.

Microsoft Power Apps 2022a – 2022d. Kuvakaappaus Microsoft Power Apps.

Microsoft Power Automate 2022a – 2022j. Kuvakaappaus Microsoft Power Automate.

Microsoft s.a. Overview. Luettavissa: <https://mvp.microsoft.com/en-US/Overview>. Luettu: 30.11.2022.

Muzyka, B. 2022. PoC vs Prototype vs MVP: What's the difference? How to choose? Luettavissa: <https://www.techmagic.co/blog/poc-vs-prototype-vs-mvp/>. Luettu: 27.9.2022.

Nduati, J. 2021. Integration Challenges and Solutions in Software Development. Luettavissa: <https://www.section.io/engineering-education/integration-challenges-and-solutions-in-software-development/>. Luettu: 4.9.2022.

Neumann, S. 2014. 8 Data Integration Best Practices. Luettavissa: <https://www.integrate.io/blog/eight-best-practices-for-data-integration-development/>. Luettu: 3.9.2022.

Niiranen, J. 2020. ALM for low-code: are we there yet? Luettavissa: <https://jukkaniirani.com/2020/06/alm-for-low-code/>. Luettu: 16.10.2022.

Norambuena, P. 2020. The 10 most common naming mistakes. Luettavissa: <https://medium.com/interbrand-australia/the-10-most-common-naming-mistakes-2020-edition-9f4b0bb97585>. Luettu: 4.10.2022.

Oracle PL/SQL developer 2022. Kuvakaappaus Oracle PL/SQL developer.

Oracle s.a. What is a Relational Database (RDBMS)? Luettavissa: <https://www.oracle.com/database/what-is-a-relational-database/>. Luettu: 30.11.2022.

Pertilä, T. 2021. Power Platform – Tietolähteiden hallinta ympäristömuuttujien avulla. Luettavissa: <https://timopertila.com/2021/04/26/power-platform-tietolahteiden-hallinta-ymparistomuuttujien-avulla/>. Luettu: 25.10.2022.

Pertilä, T. 2022. Power Apps and Git version control. Luettavissa: <https://forwardforever.com/power-apps-and-git-version-control/>. Luettu: 12.9.2022.

Pratt, M. 2020. Proof of concept (POC). Luettavissa: <https://www.techtarget.com/searchcio/definition/proof-of-concept-POC>. Luettu: 27.9.2022.

Projector 2022. Streamline How You Invoice for Professional Services: Tips and Best Practices. Luettavissa: <https://www.projectorpsa.com/blog/invoicing-for-professional-services/>. Luettu: 4.10.2022.

Ray, S. 5 Ways to Engage End Users. Luettavissa: <https://www.projectmanager.com/blog/ways-engage-end-users>. Luettu: 4.10.2022.

Robinson, S. 2020. How customer transparency helps companies maintain loyalty. Luettavissa: <https://www.techtarget.com/searchcustomerexperience/tip/How-customer-transparency-helps-companies-maintain-loyalty>. Luettu: 5.10.2022.

Schwaber, K. & Sutherland, J. 2020. Scrum-opas. Luettavissa: <https://scrum-guides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Finnish.pdf>. Luettu: 11.9.2022.

Skyline consultants s.a. PowerApps Disadvantages: When not to use PowerApps. Luettavissa: <https://skylineconsultants.com/when-not-to-use-powerapps/>. Luettu: 11.10.2022.

Shufeldt, J. 2021. Start Your Business With An MVP. Luettavissa: <https://www.forbes.com/sites/forbesbooksauthors/2021/09/13/start-your-business-with-an-mvp/>. Luettu: 23.11.2022.

Valjas 2019. Mitä integraatio, rajapinta ja api tarkoittavat? Luettavissa: <https://valjas.fi/opi/blogi/mita-integraatio-rajapinta-ja-api-tarkoittavat/>. Luettu: 30.11.2022.

Vossen, J. 2018. Software development from a business perspective. Luettavissa: <https://medium.com/spartner/software-development-from-a-business-perspective-3a02cb49cf5e>. Luettu: 10.10.2022.

W3schools s.a. XML Soap. Luettavissa: https://www.w3schools.com/xml/xml_soap.asp. Luettu: 23.11.2022.

Walnut 2021. How to Write a Killer Follow-up Email After a Product Demo. Luettavissa: <https://www.walnut.io/post/how-to-write-a-killer-demo-follow-up-email>. Luettu: 4.10.2022.

Wolford, B. s.a. What is GDPR, the EU's new data protection law? Luettavissa: <https://gdpr.eu/what-is-gdpr/?cn-reloaded=1>. Luettu: 25.9.2022.

Wong, J., Iijima, K., Leow, A., Jain, A. & Vincent, P. Magic Quadrant for Enterprise Low-Code Application Platforms. Luettavissa: <https://www.gartner.com/doc/reprints?id=1-27IIP-KYV&ct=210923&st=sb>. Luettu: 23.8.2022

xkcd.com s.a. Automation. Luettavissa: <https://xkcd.com/1319/>. Luettu: 16.10.2022.

Young, S. 18.1.2021. Power Apps Date Time Math - Do calculations for payroll and timesheets. Video. Katsottavissa: <https://www.youtube.com/watch?v=FbdGLVCvgRo>. Katsottu: 8.9.2022.